

开源时代

OPEN SOURCE TIMES

2010年3月刊 总第十八期

开源业界:

联想联合红帽在美推出服务器打折计划
Twitter或因用户暴增用Cassandra替换MySQL

行业观察:

技术新知:

利用U盘安装Linux操作系统
linux上防火墙tcp_wrappers详解

甲骨文收购Sun消化不良 Sun赢利困难
开源软件不会比“闭源”软件安全?

社区扫描:

Fedora对比Ubuntu 六大要素决胜负
Symbian操作系统提前完全开源计划

本期推荐:

开源：移动操作系统市场的搅局者？

一年一度的巴塞罗那移动世界大会在春节期间已经落幕。各大移动设备厂商、运营商和软件开发商纷纷亮相，除了层出不穷的新设备和新技术外，移动设备操作系统已经成为兵家必争之地。眼瞧着Android操作系统在美国市场攻城略地，直追苹果，Nokia再也按耐不住，联手Intel正式推出开源MeeGo操作系统。

内容目录

刊首语.....	4
开源业界.....	5
Adobe 等四厂商加入 LiMo 手机平台阵营.....	5
联想联合红帽在美推出服务器打折计划.....	6
如何平衡 Linux 与 Windows 的使用比例?	6
Linux 创始人博客: Nexus One 是个赢家.....	8
微软和亚马孙达成包括 Linux 在内广泛交叉授权协议.....	9
MWC2010: 诺基亚联合英特尔推出 MeeGo 操作系统.....	9
思杰 Novell 达成云计算虚拟化战略关系.....	10
Twitter 或因用户暴增用 Cassandra 替换 MySQL.....	10
大企业在后端使用 Linux 的十大常见方式.....	11
VMware Vs 红帽: 下一个虚拟机管理战场.....	13
Ubuntu Linux 入选美政府采购单 死磕红帽 Novell.....	14
Firefox 和 Chrome 愈战愈勇 IE 份额跌破 65%.....	15
风河打破开源魔咒.....	15
谷歌 Android 已成大量不兼容设备的起点.....	16
开源 CRM 软件商 SugarCRM 创下销售纪录.....	18
社区扫描.....	20
Fedora 对比 Ubuntu 六大要素决胜负.....	20
SRS 实验室成为首家加入 LiMo Foundation 的音频公司.....	21
Linux 社区为何驱逐 Google Android?	22
10 亿欧元才能开发出目前的 Linux 内核.....	22
发行版发布: PC-BSD 8.0.....	23
你在使用开源软件就是用盗版?	23
Novell 和 LPI 开展培训和认证合作.....	23
Apache Http Server 的 15 岁生日.....	24
自由软件基金会: Google 应该开源 VP8 编码器.....	25
Symbian 操作系统提前完全开源计划.....	25
行业观察.....	27
Google 店大欺客 伪开源 Android 危机四伏.....	27
甲骨文收购 Sun 消化不良 Sun 赢利困难.....	28
Symbian 开放源代码 开源是否等于成功?	29
开源软件不会比“闭源”软件安全?	30
本期推荐.....	31
开源: “移动航母”的核动力.....	31
英特尔的 MeeGo+X86 能否撼动 Android+ARM?	32
Linux 操作系统新贵 MeeGo 的前世与今生.....	33
技术新知.....	36

Linux 大史记---缅怀已逝的十八年.....	36
Linux 上防火墙 tcp_wrappers 详解	46
PHP 应用提高篇.....	48
利用 U 盘安装 Linux 操作系统.....	53
Python 在数据库测试中的应用探索.....	59
突破 Linux 内核模块认证.....	64
1、为什么要突破模块验证.....	64
2、内核是怎么实现的.....	66
3、怎样去突破.....	71
4、总结.....	75
5、参考.....	75
git 几种应用模式	75
Kernel 花哨的编译连接技术	79
网友热评.....	83

本期编辑：周荣茂 覃里

美工：沈正刚

编校：王文冰

投稿邮箱：rmzhou@staff.chinaunix.net

本刊网址：<http://linux.chinaunix.net/ebook/>

刊首语

新的一期杂志又要发布了，大家或许会问：2010 年《开源时代》是否会有所改变呢？

答案是：肯定的。在内容方面，我们将加强技术文章的内容量，同时将举办更多的线上或线下活动来丰富读者交流。

《开源时代》读者有奖调查活动目前已经结束，在 2 个月的时间里，有 2000 多位来自全国各地的网友参与了调查，有效反馈多达 1700 多条。说实话，看到用户反馈意见，我非常感动，不仅仅是因为反馈中更多的是肯定与鼓励，更多是网友用心撰写的多达上百字的建议。总结归纳下来，网友建议集中在如下 5 条：1、加强技术内容量。2、期待出纸质杂志。3、期待更多的线上线下活动。4、加强杂志宣传。5、提供更好的阅读体验。

首先需要谈的是加强技术内容的问题，这个问题确实非常重要，其实早在建刊之初《开源时代》定位就是一本以技术为核心的刊物，但需要坦诚的是，因为我们的杂志是免费的，靠稿费是杯水车薪，因此今年我们会启动一系列有奖征文活动，来丰富我们技术文章的同时也尽量给投稿的读者提供物质激励，当然这还需要大家积极来参与。

关于出版纸质杂志的问题，其实我们也很期待，因为这也一直是我们的目标，不过目前时机还未成熟。大家都知道，其实报纸也好，杂志也好，要想生存下去，不是靠读者订阅的盈利，而是企业广告赞助。没有赞助，印刷厂可不能“开源”印刷。因此这方面短期内还无法实现。

阅读体验方面，我们从本期开始提供下载的同时将提供本期杂志的内容目录，这样读者在选择的时候会比较灵活。另外关于活动，2010 年，我们将会有更多的活动与大家见面，4 月 2 日，2010 数据库技术大会即将召开，欢迎大家参与。杂志的宣传更需要依靠广大读者的口碑传播，让我们在 2010 年一起努力，一起呵护目前国内剩下的唯一一本开源免费读物！

----IT168 资深编辑：覃里

开源业界

Adobe 等四厂商加入 LiMo 手机平台阵营

开源手机操作系统开发机构 LiMo 基金会近日宣布，美国知名软件开发商 Adobe 及其他三家科技公司已正式成为 LiMo 基金会新成员，从而使 LiMo 平台的影响力进一步扩大。

LiMo 基金会成立于 2007 年 1 月，是一家全球性行业组织，并在英国伦敦和日本东京设有总部。其名称为“Linux Mobile”(Linux 移动)的英文缩写。LiMo 基金会成员包括全球手机制造商、移动运营商、手机芯片制造商及其他相关企业。LiMo 基金会的宗旨是：加强各成员之间的合作力度，以共同开发基于 Linux 的开源手机操作系统。

就目前全球手机平台而言，当以诺基亚的 Symbian 平台领先。但过去一年中，苹果 iPhone 和加拿大智能手机制造商 RIM 的黑莓(BlackBerry)手机抢走了不少原本属于 Symbian 的市场份额。与此同时，基于 Linux 内核的手机平台市场份额正呈不断增长之势。除 LiMo 平台外，谷歌 Android 手机平台也是基于 Linux 内核。



LiMo 基金会执行主管摩根·吉利斯(Morgan Gillis)在接受外界采访时表示：“Linux 手机平台正向前迈出重大步伐，任何其他手机平台的设备涵盖范围都无法同 Linux 手机平台相比。事实上，所有主流手机制造商都在推出基于 Linux 的智能手机。”

虽然 LiMo 和谷歌 Android 同为开源平台，且市场目标颇为一致，但二者具体的技术开发手段和推广策略却大为不同。LiMo 着重发展其成员数量，然后再通过成员之手把 LiMo 手机推向市场。Android 则更愿意吸引大量外部开发者加入 Android 平台，使他们愿意为 Android 平台开发各类应用软件。

吉利斯去年 10 月曾表示，不会效仿谷歌开通自家应用程序商店，但此前已向外部开发者提供了相应开发工具，并计划今年初对这些工具进行升级。也就是说，LiMo 用户今后可到移动运营商或其他服务商的应用程序商店下载相关软件。从这个角度上讲，LiMo 在应用程序使用上的开放性和灵活性程度将更高。

联想联合红帽在美推出服务器打折计划

这次联想推出的打折计划是由三方推广的，联想负责服务器，红帽提供操作系统服务支持，Tech Data 作为分销商。

这个服务器打包打折计划看似简单，但是却有玄机。其中有三点值得我们探讨：1、开源技术已经深入 IT 渠道销售；2、美国大的分销商也开始推广开源技术；3、联想的服务器市场策略。

总的来说，开源技术的 IT 渠道不断得到拓展，这是业界相关调查已经得到的结果，虽然目前做的不好，还有很多的可以发展的空间，但是可以证明，开源技术已经开始被企业和消费者接受，比如 Linux 服务器虚拟化技术，Firefox 桌面软件等。

开源软件在小型商业公司并没有得到很好的应用，这里更多的是微软的天下，但是现在也有越来越多的小型商业企业也开始关注 Linux 及其他的开源技术软件，毕竟他们在大中型企业中的影响力越来越大。

分销商的进一步介入

同时，类似于 Tech Data 的分销商，比如他的竞争对手 Synnex，也已经开始向他们的合作伙伴和客户推广他们的开源解决方案。Synnex 和红帽去年年初的时间就推出了开源渠道联盟的市场策略。Tech Data 稍晚一些，在 2009 年底的时候联合 Open Tech 推出。

但是目前看来，Tech Data 和 Synnex 的效果并不是很明显，但至少看到一点，那就是大家已经开始行动。效果不明显的原因可能是这样的，这些大的渠道商下面的二级分销商并不能一下子消化打包推出的开源解决方案，因为很多的二级分销商还停留在对开源的理解就是 Linux 的阶段。现在的解决方案很多都是包括开源的企业级应用软件，比如开源数据库产品、开源 CRM、开源 ERP 和开源 CMS。他们对这些开源企业级软件的分销和售后经验不足。

联想的下一步

作为这个联合打包打折计划，联想也想为他的服务器产品线打开一个蓝海。联想从 2008 年的开启了他的 ThinkServer 服务器产品线，当时该服务器搭配 Windows Server 和 Novell SUSE Enterprise Linux 操作系统。

自从那个时候起，联想也开始提供对红帽 RHEL 产品的支持。目前这一打包打折方案的推出，也证明了联想和红帽想进一步把 Linux 推向中小企业市场。而分销商 Tech Data 的加入，也为他们帮了一个忙。

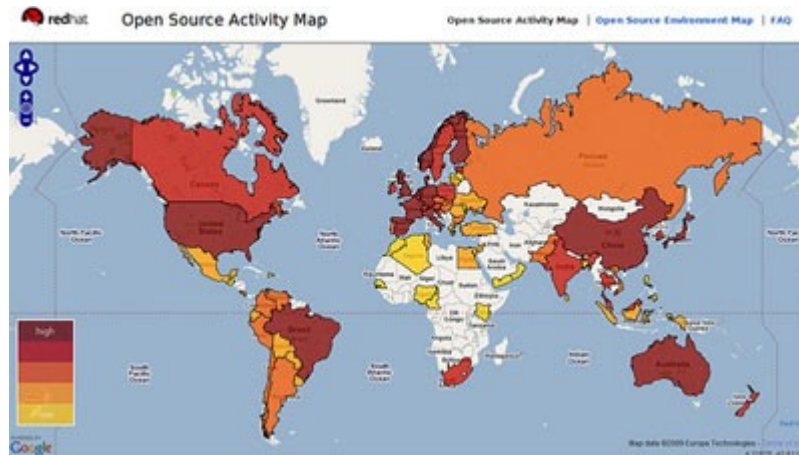
联想作为硬件厂商，它需要一个强大的服务器产品线，否则它无法和 HP、Dell 和 IBM 展开全面竞争。而此同时，红帽需要一个强大的合作伙伴来帮助他在中小企业中扩大影响力。所以这次三方合作看起来很普通，但是细细想来这可能会对开源技术行业产生深远影响。

如何平衡 Linux 与 Windows 的使用比例？

企业要能顺应时代的潮流使用开源软件、又不能把现有的 MS Windows 系统舍弃掉，难道真的是要每一位员工都在已安装 MS Windows 的个人电脑上再安装一套 Linux，成为双操作系统吗？

目前看来是不大可能的。Linux 与 MS Windows 的竞争诉求主要是在于软件的开放与专属，两种操作系统都可以独立运作，其基本功能包含视窗人机介面、网络、数据库连接等相关衍生功能都是类似而不是互补的，所以装在一部电脑上不会同时运作。

一般企业里多数忙碌的上班族在用电脑工作的时候，讲求的是方便与效率，因此无谓的键盘选择或鼠标移动等动作是越少越好；如果觉得 MS Windows 不适用就会考虑改用 Linux(反之亦然)，不会在两个操作系统之间不断切换，增加不必要的动作。至于以后会变成什么样子，是不是微软也出一套开源的操作系统(叫做 Winux?)，或是 lindows 冒出头来，说实在谁也没把握，但我想最后应该不会是两个操作系统装在一部个人电脑上。



因此，笔者认为平衡企业内 Linux 与 MS Windows 的应用比例，要先知道服务器、个人电脑等的数量，再分别统计所安装的不同操作系统数目为依据才比较客观。如果操作系统的总数大于电脑的总数，那不管比例如何，所谓的平衡是可质疑的。

“平衡”的主要意义在于合理化而不是均分化，否则企业只要把电脑总数除以二，一半用 Linux，另一半用 MS Windows 就行了。分散式(Distributed)与集中式(Centralized)软件架构选择的例子可以说明平衡的意思。

早期的信息系统是只用一部大型电脑来担纲，简单明了，但如果电脑宕机，整个系统就会停摆。后来，就有人提倡用好几部小一点的电脑以网络连起来共同合作，可以达成原来大型电脑的系统功能，虽然比较复杂，但如果其中有一部电脑宕机，其他的电脑还能保持继续运作，功能虽有减少，但不致造成系统全面停摆。集中式系统与分散式系统当然有其历史背景，但是分散式系统并没有把集中式系统淘汰掉，因为集中与分散各有其架构上的优点；前者适于要有全面掌握性的应用，后者适于工作可以分摊处理的应用。一般企业资源规划系统都有一个集中式的数据库来掌控全盘的经营资讯，但是并不是把所有部门的资料全部都放在里面；各部门有其独立的数据库，而集中式数据库里有指标说明如何到各分散的数据库里去找出有关的资料。

企业在决定各个应用系统的架构区分时，并不是以应用系统的数量为因素，而是以应用的性质为基础，譬如，属于全盘管理、控制类的应用功能要采集中式架构来设计；属于各别计算、处理资料类的应用功能则可以采用分散式架构来设计。平衡企业内 Linux 与 MS Windows 的应用比例，也要以应用性质为基础。虽然两种操作系统都可以担任服务器，通常我都会建议把有多个工作(task)要同时处理、互动、管控类的隐性应用(例如网络监测、资料存取权限管理等使用者不会直接感觉到的功能)交由 Linux 负责；把比较靠近使用者、需要亲和力的显性应用，交由 MS Windows 负责。至于使用者端的个人电脑或笔记型电脑，我看短期内大概还都是用 MS Windows，不必勉强改用 Linux 会比较讨好。

到底要怎样平衡呢？再从实际应用面来观察一下也许会更明白。

企业用 Linux 建置电子邮件服务器、网站服务器等等已经非常普遍，用 Linux 来担任基本网络安全的防火墙也很称职，那剩下来的部分主要就是数据库服务器与应用服务器了。全世界一些著名的

数据库管理系统为了市场竞争，大多提供高于业界共同标准功能的应用软件呼叫界面，但这些介面却不相容以致于相关应用软件，无法轻松换用不同的数据库管理系统，虽然他们都是合乎 SQL 标准。因此，数据库服务器是否要采用 MS Windows 或 Linux 必须看所采用的数据库管理系统是否支持该操作系统。

我的了解是这些数据库供应厂商都已经宣布他们支持 Linux。应用服务器最为复杂，因为应用部份本来就缺乏标准，不管底层用 Linux 或 MS Windows，上层没有应用软件是没有意义的。微软的绝对优势是就在 MS Windows 之上把应用软件开发出来，不管有没有标准，反正在整个 MS Windows 环境下应用软件可以执行就是了。相对地，Linux 的条件就比较吃亏，还待努力。我对应用服务器的看法是，企业里每个人都会用到的应用系统，(譬如门禁、电子表单)不妨考虑采用 Linux 及 MySQL；支持管理部门专用应用系统(譬如人事、财务)也许就用 MS Windows 会大家欢喜。

再从另外的角度来看，平衡的作法也可以只对 Linux 或 MS Windows 的价格、相关技术服务水准、原厂的承诺、可能风险等企业管理的因素着手。笔者认为 Linux(或 MS Windows 亦同)的比例若是占整个企业所有电脑数目的 1/3 至 2/3 之间，应该都是属于不错的平衡态势，至少管理人员心理上感觉比较安全。如果不在这个范围内，或是根本算不清楚，那可能要再多费点心。

Linux 创始人博客：Nexus One 是个赢家

Linux 核心程序开发计划的领导者 Linus Torvald，在周六对 Google 手机大加表扬，说他很喜欢这款 Google 手机，还说这足以让他变成爱上手机。



Torvalds 在博客中说：“我很讨厌电话，他们在你工作或是阅读或是其它随便什么时候干扰打扰你，而手机对我来说则是一个可以在任何地点被打扰的机会，但是我得承认，Nexus One 是个赢家。”讲电话这件事情并没有让 Torvalds 很兴奋，而是其它功能让他心动。

文章称：“无论如何我想要在车上有一台 GPS 装置，而 Google 的导航可能终于让一台手机有了用处。而它真的是如此，这真的很不一样。我再也不需要身上拖着一支手机只为了以防我需要和某人联络，现在我有一款有用（而且真的还蛮好看）的玩意。而你可以拿它来讲电话这件事情仅是其次。”

Nexus One 里面使用的 Google Android 操作系统是以 Linux 为基础，但是一般来说程序并不是执行在 Linux 上。而是执行在 Google Dalvik 虚拟机器和伴随的软件数据库，一个类似 Java 的

表层上。不过最近 Google 发行一个原生开发套件 (Native Development Kit) 让程序可以直接跑在该手机的 Linux 操作系统上。透过这项技术, Mozilla 正在开发 Android 版的 Firefox。

Torvalds 写道: “在 Android 上工作目前来说算蛮好的, 因为它的 Java 核心, 所以对我们来说是个有点不同的平台, 但是透过 NDK 我们可以把东西桥接在一起不会遇到什么问题。” Mozilla 正在开发这款软件的 Vladimir Vukicevic 在周四说: “距离可用的 alpha 版释出还有段距离, 但是我们的确更进一步了。我们现在有了基本架构工具, 也有办法加速我们的进度了。”

微软和亚马孙达成包括 Linux 在内广泛交叉授权协议

近日, 据国外媒体今日报道, 微软表示, 已经与亚马孙达成了广泛的交叉专利授权协议。

微软表示, 除了允许亚马孙使用基于 Linux 的服务器外, 这一协议还涵盖了亚马孙 Kindle, 包括该电子阅读器所使用的开源和专有技术。虽然双方均未透露这一协议的财务细节, 不过微软表示, 亚马孙将向其支付一笔资金, 但并未透露具体金额。

微软此前曾经表示, Linux 及其开源技术侵犯了该公司的专利。微软去年曾经起诉导航服务公司 Tom-Tom 在设备中使用了 Linux 内核。

该消息由微软发布, 但其中并不包含亚马孙的声明。

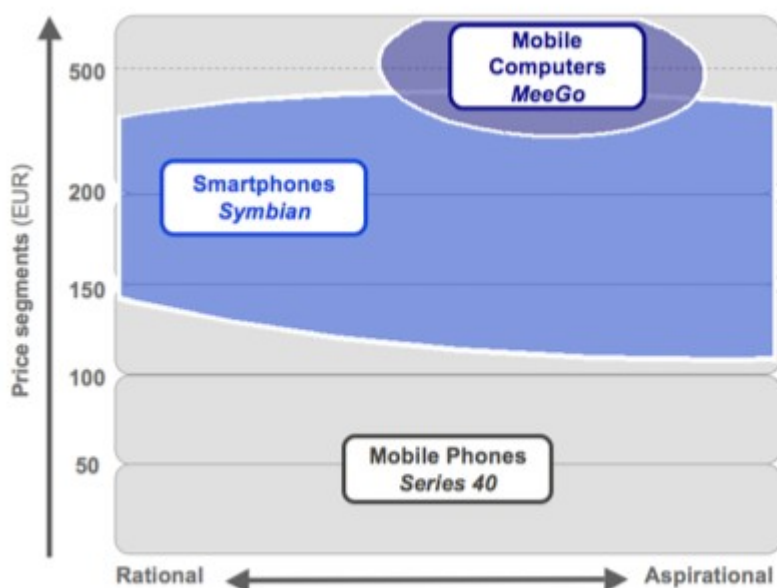
MWC2010: 诺基亚联合英特尔推出 MeeGo 操作系统

2010 年 2 月 15 日, 在本年度 MWC 大会的联合发布会上, 诺基亚和英特尔正式对外宣布将联手推出新的移动操作系统——MeeGo。MeeGo 将融合两个公司现有的移动操作系统, 采用 Linux 架构, 并可以扩展到包括智能手机、上网本、联网 TV、以及 desktop 电脑上。



据透露, 新的操作系统将融合两个公司现有系统的优点, 包括 Moblin 的代码和 Maemo 的用户界面, Moblin 是英特尔开发出来的, Maemo 属于诺基亚公司。第一版的 MeeGo 将在第二季度面世。英特尔软件和服务集团的副总裁 Renée James 宣称, 第一台基于 MeeGo 系统的设备预计将于 2010 年下半年研发成功。新的系统将支持英特尔的 Atom 处理器和 ARM 架构。MeeGo 的源代码及开发者工具将于近几周对外公布, 届时大家可在其官网 <http://meego.com/> 上查询、获取。

MeeGo 将建立在 Linux 的基础之上, 任何人都可以在 Qt 架构的基础上为 MeeGo 开发程序, 一旦开发完毕, 就可以在配置 MeeGo 的各个平台上使用。到目前为止, Nokia 是唯一一家明确表示将推出 MeeGo 手机的公司。不过, 相信在几个星期之内, 就会有越来越多的硬件厂商和开发者会推出自己相应的 MeeGo 计划, Renée James 宣称。



据 nokia 透露，MeeGo 的发布将不会影响到现有的 Symbian 系统，Symbian 将会被继续用在诺基亚低端的智能手机上。据业内人士分析，MeeGo 发布后，诺基亚的移动操作系统战略将如上图所示。

思杰 Novell 达成云计算虚拟化战略关系

思杰系统公司和 Novell 公司本周表示，他们将在评估工具的互操作性和协作性上进行合作，以降低管理成本和 Linux 和 Windows 操作系统上运行的多个虚拟服务器的管理复杂性。

这项新伙伴关系的确立，使 Novell 认定在 Citrix XenServer 的虚拟化套件运行"的 SUSE Linux 企业服务器的用户将逐渐增多，使其成为唯一可以凭借 Linux 操作系统来与微软，VMware 等软件公司来抗衡的公司。

该公司官员说，超过 4500 个企业在已选定的"Novell SUSE Linux 企业服务器"认证应用程序中准备部署思杰的产品，计划在 SUSE Linux Enterprise 服务器上运行 XenServer 虚拟机。

"我们与思杰合作，为我们许多共同的客户和合作伙伴提供了更好的方法来管理他们的虚拟基础架构，" Novell 公司全球联盟的总经理 Joe Wagner 在声明中说。"我们的企业提供一个更深刻和全面的综合设置解决方案，为数据中心虚拟化提供更加明智的解决方案，以及世界一流水准的运行高性能计算的虚拟化环境。"

思杰和其他虚拟化和云计算支柱型厂商都在追求更加广泛的合作平台策略，以此来增加自身的竞争力。由于迅猛发展的虚拟化市场而导致这种爆发性的厂商间合作。例如去年 11 月，VMware 公司宣布与思科系统公司及 EMC 公司达成 VCE 联盟，帮助企业简化并加速他们的数据中心虚拟化。

VCE 联盟的目的还在于发展私有云架构，并借此在多种操作系统上运行不同的应用程序和业务。

Twitter 或因用户暴增用 Cassandra 替换 MySQL

据外电报道，Twitter 公司一位名叫 Ryan King 的工程师日前向博客 MyNoSQL 透露，公司计划从 MySQL 迁移到 Cassandra 数据库，因为后者具有更大的弹性、可扩展性和大量的社区网络开源

开发人员。“我们有大量的数据，在数据巨大，增长率正在加速的情况下，我们需要一个系统，它可以更为自动化，并高度可靠、可用。” Ryan King 表示。相关数据显示，在 2009 年里，Twitter 用户数从 1 月份的每天约 200 万人，激增至 12 月份的每天近 5000 万人。每天登陆 Twitter 的用户数量增长已经超过了 20 倍。



事实上，对于熟练的程序员来说，所谓的 NoSQL 数据库可以提高性能和可扩展性，这使得它们吸引消费者的网络世界。像 Facebook 和 Twitter，以及最初的 Web 2.0 的宠儿——Digg，也正从 MySQL 数据库迁移到 Cassandra 数据库。

Ryan King 表示，试验发现，Cassandra 数据库比 MySQL 更具有可扩展性、可靠和更易于管理。“把装有所有 tweets 和 retweets 信息的平台迁移到 Cassandra 上去，这是 Twitter 的当前要务。在这之后，我们将开始把一些新的项目迁移到 Cassandra 或者其他平台。”

大企业在后端使用 Linux 的十大常见方式

世界上最大的技术支持、软件和硬件公司每天使用 Linux 完成各种任务与解决方案，那么这些大公司究竟是怎么使用 Linux 的呢？其实并不神秘，本文为你揭晓答案，大多数公司都不会使用 Linux 作为桌面操作系统，主要是用于后端服务器操作系统，经过这些大公司的大胆尝试，许多事实证明 Linux 完全可以担负起关键任务计算应用，并且有很多 Linux 系统从开始运行至今从未宕过机，100% 的正常运行时间让人无不惊叹，当然你也可以做到。

10、入侵检测系统

Linux 天生就是一个完美的入侵检测服务平台，因为它是免费的，且可以运行在很多种硬件平台上，同时也是开源爱好者喜欢的平台，Linux 上最著名的入侵防御和检测系统要数 Snort，它也是开源且免费的。

下面的话引自 Snort 网站：

Snort®是由 Sourcefire 开发的开源网络入侵防御/检测系统(IDS/IPS)，结合了签名、协议和基于异常的检测，Snort 是世界上部署最广泛的 IDS/IPS，数以百万计的下载量和超过 270,000 位注册用户，Snort 已经成为事实上的 IPS 标准。

如果你还没有使用 Snort，你应该尝试一下。

9、Google 搜索设备

Google 在 Linux 平台上构建起搜索设备，如果你的公司在使用这种设备，那么你就在使用 Linux，但 Google 使用的 Linux 非常特殊，专门进行了定制和优化。

8、监控服务

如果你要做网络监控或系统性能监测，那么 Linux 是一个不错的选择，大公司一般使用淘汰下来的硬件设备和自由软件搭建监控系统，如 Orca 和 Sysstat 都是 Linux 上不错的监控方案，IT 专业人

员利用它们可以实现自动化监控，无论你的网络是大是小，它们都能应付自如。

7、开发平台

Linux 下有许多开发工具，如 Eclipse、C、C++、Mono、Python、Perl、PHP 等，毫无疑问，Linux 是世界上最流行的开发平台，它包含了成千上万的免费开发软件，这对于全球开发者都是一个好消息。

6、日志服务器

Linux 是处理和存储日志文件的绝佳平台，听起来这是一个低级的任务，但它的低成本，低硬件要求，和高性能是任何需要日志服务的人的首选平台，大公司也经常使用 Linux 作为日志服务的低成本平台。

5、跳转盒(Jump box)

对于企业而言，跳转盒是一个为公共网络(如互联网)到安全网络(如客户部)提供的网关，这样一个廉价的系统也可以为大量的用户提供服务，而相对应的 Windows 系统需要成千上万美元的终端服务访问许可和客户端访问许可，并且对硬件的要求更高。

4、应用服务器

Tomcat, Geronimo, WebSphere 和 WebLogic 都是 Java 应用服务器，Linux 为这些服务提供了一个稳定的，内存消耗很小的，可长时间运行的平台。IBM 和 Oracle 也都非常支持 Linux，它们也逐渐将 Linux 作为其软件系统的首要运行平台。

3、Web 服务器

我想现在任何人都知道 Apache 是世界上用的最多的 Web 服务器吧，至少最近 10 年是大家公认的事实上的 Web 服务器标准，那么它运行在什么平台上呢？答案是所有的平台都支持，但超过 90% 的 Apache 都是搭配 Linux 运行的。

2、数据库服务器

Oracle 和 IBM 都有企业级软件运行在 Linux 上，为什么？因为它们在 Linux 上可以工作得很好，Linux 自身消耗的资源很少，因此它不会和数据库进行资源的抢夺，一个 RDBMS 需要一个稳定的，无内存泄露的，快速磁盘 I/O 和无 CPU 竞争的操作系统，Linux 就是这样的系统，世界上已经有很多开发人员使用 LAMP(Linux, Apache, MySQL 和 Perl/PHP/Python)和 LAPP(Linux, Apache, PostgreSQL, Perl/PHP/Python)作为开发平台，也有很多关键应用系统是这么部署的。

1、虚拟化

从桌面虚拟化到云，现在又回到桌面虚拟化，VMware 是虚拟化产品做得最早也是目前最好的一家公司，现在它的主要产品也是基于 Linux 的，另外 Citrix, Red Hat 以及微软也是 VMware 的有力竞争者。

总结

通过上面 10 种 Linux 使用方式，相信你不会再觉得 Linux 是在夹缝中求生存，Linux 也不只是为桌面，家庭用户及小公司而存在的，它在大公司中同样受到重用。你在你的公司是怎么使用 Linux 的呢？是个实验性的平台，还是一台 DIY 路由器，防火墙？欢迎发表你的意见和想法！

VMware Vs 红帽：下一个虚拟机管理战场

红帽公司计划加强旗下虚拟化管理软件的功能，以备他们将来能和 VMware 站在更加势均力敌的层面上展开竞争。

Linux 供应商红帽公司去年 11 月推出了针对服务器裸机管理程序解决方案的红帽企业版虚拟化软件。与内置 KVM 虚拟化程序的红帽企业版 Linux(RHEL) 5.4 的推出时间仅仅相隔两个月。服务器裸机管理程序解决方案由两部分组成：以 KVM 为基础的管理程序本身（这也是 RHEL 核心的基础）和服务于裸机管理程序解决方案管理器（RHEV Manager for Servers）。这款管理器是一款在 Windows Server 2003 操作系统环境下运行的虚拟化管理系统，可以帮助管理员控制在服务器裸机管理程序解决方案或 RHEL-WITH-KVM 主机上运行的虚拟机并自行托管。



很多业内人士将 VMware 目前在虚拟化领域占据的统治地位归功于 VMware 产品强大的管理基础架构，相对还不够成熟的红帽服务器裸机管理程序解决方案管理器在高端领域还无法与 VMware 管理系统的品质相匹敌。不过这款软件也包括了高可用性，系统日程安排程序和存储管理等特性，其中还有一项非常重要的特点不能不提：那就是实时迁移能力（VMware 将其称为 VMotion）。

实时迁移就是当虚拟硬盘文件保留在共享存储磁盘上时，将运行中的虚拟机在无需中断系统运行的情况下就能从一台主机迁移到另一台主机的能力。微软的 Hyper-V 管理程序当初推出时遭遇批评就是因为它缺少实时迁移的特性，因此红帽公司在推出虚拟化软件时将这项功能囊括其中也在意料之中。

VMware 具备而红帽服务器裸机管理程序解决方案管理器缺乏的两项特性是执行虚拟磁盘文件实时迁移的能力（VMware 将其称为 Storage VMotion）和类似于 VMware Consolidated Backup 的功能。但是这也是红帽计划解决的问题。红帽公司虚拟化事业部高级总监 Navin Thadani 表示“我们目前还不具备 storage VMotion 功能，但是我们正在开发这项特性，将来这一切都会改变”。在提及和 VCB 抗衡的产品时，他表示目前还没有这个必要，因为多数用户更愿意用他们备份物理硬盘的方式来备份虚拟硬盘，即使用在虚拟机上运行的备份代理。不过虚拟机上运行的代理在备份期间会耗费物理主机的资源，随着虚拟机使用率的攀升，红帽的用户很可能对无代理解决方案有所需求。Thadani 承诺说“我们正在和公司的合作伙伴进行磋商，未来我们将设计类似 VCB 的功能。用户在红帽和 VMware 之间进行选择时，将是旗鼓相当的特性比较”。

KVM 的优势

Thadani 还认为红帽以 KVM 为基础的虚拟化产品目前在两方面具备超越 VMware 的优势--即

价格和应用软件支持。精确的成本计算取决于主机的数量和服务器及相关操作系统虚拟化的数量。不过，如果使用红帽服务器裸机管理程序解决方案来替代 VMware 管理程序将帮助用户节约大概 40%到 80%的费用。Thadani 宣称红帽公司将向用户做出保证，因为红帽服务器裸机管理程序解决方案和 RHEL 一样是采用同样的核心搭建的，任何在 RHEL 上运行的 Linux 应用软件也可以在红帽服务器裸机管理程序上的 RHEL 虚拟机上运行。如果用这种方法不能正常运行的话，红帽将提供修正支持来帮助用户完成运行。



Thadani 宣称，红帽可以与 VMware 在 VMware 平台上运行 RHEL 客户机的企业用户领域进行竞争。这些企业中有些在得到红帽的无限量客户端许可证授权后，可以将 RHEL 客户端在运行 RHEL-WITH-KVM 的主机上对其虚拟化。Thadani 认为，红帽目前可以达到像 SAP，甲骨文应用软件和虚拟化环境下运行的 DB2 软件等产品大概 85%到 95%的性能级别。用户也会在红帽服务器裸机管理程序解决方案上同时运行 Windows 和 RHEL 客户端，这么做的目的就是减少他们对 VMware 的依赖。“用户正在寻找 VMware 的替代选择，他们会发现红帽服务器裸机管理程序解决方案是值得信赖的。用户需要除了 VMware 之外的第二选择”。

红帽还计划推出诸如 Storage Live Migration 等特性并将红帽服务器裸机管理程序解决方案管理组件的下一代版本的备份特性整合在一起，新的功能将以 RHEL6 核心（RHEL6 核心可能是在今年的某个时候推出）为基础来为管理程序设计。Thadani 暗指 RHEL6 核心能比目前的版本帮助红帽服务器裸机管理程序更好的平衡 SELinux 的特性。Thadani 称“这将帮助我们保障管理程序的安全，让管理程序变得更加难于侵入，即使发生了安全问题，也不可能导致任何虚拟机的瘫痪”。RHEL6 中的 CFS(Completely Fair Scheduler)特性也能让管理员针对特定的虚拟机设置精确的 SLA，以此来保证中央处理器，内存或者输入/输出资源使用的最小化。而且还能让公司将大型的关键任务应用软件迁移到内部云上。

显然红帽服务器裸机管理程序解决方案和 RHEL-WITH-KVM 在和整个 VMware 生态系统形成势均力敌的竞争关系之前，还有很长的一段路要走。但是红帽公司看起来是下定决心要主推旗下的虚拟化产品，逐渐丰富产品的管理特性并缩小与 VMware 的差距，直到他们能和 VMware 并驾齐驱。

Ubuntu Linux 入选美政府采购单 死磕红帽 Novell

据外电报道，继红帽公司（Red Hat）和 Novell 之后，Canonical 公司 Ubuntu 软件通过了美国总务署（The U.S. General Services Administration）的优先采购计划，而成为第三家向美国联邦政府用户提供 Linux 操作系统的厂家。

Cole Crawford，一家 IT 服务公司 Autonomic Resources 的 CTO 表示，“如此一来，政府用户将可以使用 Ubuntu，就和用 Landscape 监控管理工具一样。”据了解，Landscape 是一款提供给 Ubuntu 服务器、桌面客户端和移动设备体积轻小的管理工具。它将成为政府客户云计算平台的一部分。

Crawford 表示，Ubuntu 是目前开发者在云计算解决方案时，所使用的最流行的操作系统。目前，开源操作系统已经被应用在一些政府机构，如 NASA。但还没有被广泛使用，是因为它缺乏 GSA 计划定价和支持。

Firefox 和 Chrome 愈战愈勇 IE 份额跌破 65%

微软 IE 的绝对优势在不断减弱，这已经是不争的事实，其竞争对手却愈战愈勇，其中，开源浏览器 Firefox 上升趋势稳定，新秀 Google Chrome 也表现不俗。根据市场调研机构 Janco Associates 公布的 2010 年 2 月份浏览器市场统计，IE 在全球的份额已经跌破 65%，在过去 4 年里，IE7 和 IE8 的发布并没有对 IE 的发展起到明显的促进作用。

Janco Associates 公布的浏览器与操作系统市场份额白皮书中指出，从 2009 年 2 月到 2010 年 2 月，IE 损失了 6.21% 的市场份额，从 70.99% 跌至 64.78%。“过去 12 个月份最大的变化是微软浏览器份额持续下跌，损失了 6% 市场。Firefox 并没有出现较大变化，Google 的 Chrome 赢得了 2.12% 市场，网景则消失了。”

在发布四个月 Windows 7 的市场份额攀升至 12%，虽然 IE8 为其默认浏览器，但是 IE 仍然继

	IE	Netscape	Firefox	Mozilla	Opera	Google
Sept 1997	49.33%	45.95%	0.00%	0.00%	0.00%	0.00%
Jan 1998	58.85%	29.38%	0.00%	0.00%	0.00%	0.00%
Jan 1999	79.44%	11.35%	0.00%	0.00%	2.06%	0.00%
Jan 2000	85.38%	5.00%	0.00%	0.00%	1.91%	0.00%
Sept 2000	70.01%	0.00%	0.00%	1.00%	0.05%	0.00%
Jan 2001	80.71%	0.74%	0.00%	13.92%	0.08%	0.00%
Jan 2002	81.66%	0.24%	0.00%	8.27%	0.17%	0.00%
Jan 2003	82.39%	0.96%	0.00%	3.86%	0.22%	0.00%
Jan 2004	80.74%	3.01%	0.00%	3.06%	0.42%	0.00%
Jan 2005	80.61%	1.74%	4.44%	2.17%	0.26%	0.00%
Jan 2006	82.10%	1.47%	11.82%	1.12%	0.50%	0.00%
Jan 2007	77.11%	1.62%	16.06%	0.96%	0.71%	1.65%
Jan 2008	71.54%	1.58%	18.61%	1.44%	0.32%	4.12%
Jan 2009	67.00%	0.54%	20.91%	2.60%	0.46%	3.95%
Aug 2009	67.98%	0.87%	19.22%	2.10%	1.10%	3.73%

© 2009 Copyright Janco Associates, Inc.

续下跌，这也就意味着一些用户并没有坚持使用 IE8。具体来看，2010 年 2 月份，Firefox 的市场份额为 17.38%，Chrome 浏览器份额增至 5.78%，苹果 Safari 和 Opera 的市场份额分别为 1.39% 和 0.41%。

风河打破开源魔咒

在全球爆发金融危机这段时期，我们观察到，也许是得益于较低的成本优势，一些开源产品销售逆势增长，这些从诞生之日就饱受质疑的开源厂商在恶劣的经济环境中表现出超强的忍耐力。多年来被诟病最多的盈利模式，优秀的开源厂商已经找到突破口。

盈利不再难以企及

开源软件企业的口袋渐渐鼓了，有的收入几百万美元，有的上亿美元。

美国 JasperSoft 业绩分析公司仅有 80 名全职员工，它的主打产品“业绩分析软件”，全球有 9 万多开发者和志愿者参与程序源代码的开发。JasperSoft 的商业模式是：基本软件免费，支持服务和增值服务收费，由于比其他商业软件的费用低很多，JasperSoft 的付费用户已经超过 9000 家。

美国风河系统公司在中国有稳定的客户群，几个月以前，它被英特尔以 8.84 亿美元收购，这是英特尔过去 8 年中首次重要收购。风河 4 年前进入 Linux，当时并不被看好，但几年发展下来，风河

Linux 业务收入已经接近 1 亿美元。市场研究机构 VDC 数据称：风河在 Linux 实时操作系统市场占有率为 32.9%，领先微软位列第一，另外它在嵌入式市场和嵌入式多核软件市场的占有率均为第一。

“风河用 4 年时间，把开放源码从一个口头的概念转变成为可以获利的商业机制，通过将 Linux 产业链上下游进行整合，使得其合作伙伴和客户都能寻找到一个基于开源的生存与发展方式。”美国风河系统公司资深副总裁兼 Linux 产品部总经理 Vincent Rerolle 接受采访时说。



紧随苹果 iPhone，Google 手机产品 Gphone 即将在中国上市，其操作系统平台 Android 是 Symbian、Windows Mobile 和苹果等手机操作系统的劲敌，其开放性、廉价性和人气，都堪比任何时代的任何手机操作系统，风河在开发 Android 中发挥了重要作用。

成功的关键

开源的商业模式基本都是“软件免费+服务收费”，强大的开发能力和很好的服务能力，两者缺一不可。Rerolle 认为，风河之所以能在很短的时间内迅速盈利，并成为嵌入式实时操作系统的领导厂商，要归功于风河一直坚持的“定期的系统化客户反馈”、“严格的产品测试”、“对重要开源组织机构的重点投资”和“全球顶级的技术支持体系”。

用户讲述了他们的体验：从风河产品中体验到开源操作系统带来的灵活性和高投资收益，通用的平台可以应用到不同的产品线，大大降低了成本。更关键的是，商用级系统支持服务非常可靠，能够有效支持企业业务的发展。这改变了开源产品因免费或廉价而服务质量差的传统形象。

眼下，风河正向更多的领域扩张。凭借服务及研发能力，它打算为 Andrors 移动手机厂商提供商用的开发平台 OEM 服务。而在被英特尔收购之前，风河与英特尔推出了面向汽车行业的全新开源 Linux 平台一车载信息娱乐系统，包括宝马、博世、Delphi 等知名厂商都加入其中。

谷歌 Android 已成大量不兼容设备的起点

现在几乎每天都有新款 Android 设备推出，乍一看，这是一件好事。在苹果 iPhone 选择独家运营、RIM 黑莓功能不够强大的情况下，硬件生产商和移动运营商非常热情地拥抱谷歌 Android 平台，将其当作智能手机操作系统的重要选择。

大量 Android 设备的出现看似是一件好事，然而事实正好相反，是一种自我毁灭的行为。越来越多的手机搭载 Android 操作系统面世，但是操作系统本身和用户所安装的应用程序均没有明确的升级策略，核心功能的配置不够连贯。简而言之，Android 平台越来越不像一个平台，而仅仅是大

量不兼容设备的起点。很少有 Android 智能手机用户愿意与移动运营商签定两年的消费协议，因为或许下个月就会有新版本的 Android 系统推出，而目前安装在智能手机上的 Android 系统和其它应用或许在未来无法得到支持。

我们可以看看去年 11 月份摩托罗拉 Droid 手机推出后出现的情况。摩托罗拉 Droid 搭载的是 Android 2.0 操作系统，宏达电的 Droid Eris 也是相同版本的平台。然而，摩托罗拉 Droid 不支持第三方应用的多点触控，但宏达电的 Droid Eris 支持这一功能，因为宏达电使用了自己的 Sense UI 界



面。与此同时，使用第一代 Android 设备(即搭载 Android 1.6 或更早版本的产品)的用户则基本上无法使用到最新的 Android 版本。

进入 2010 年，这种情况更加混乱。部分运营商将早先的 Android 智能手机更新到 2.1 版本，但很多运营商没有选择更新。2.1 版本的 Android 操作系统随着谷歌的自主品牌手机 Nexus One 面世。谷歌将 Nexus One 称为“超级手机”，并称之为 Android 平台的标准搭载设备。然而，这款手机仍然不具备多点触控，没有键盘，同时谷歌的售后服务也一团糟。下个月，Nexus One 的代工商宏达电将推出新款手机 HTC Desire。这款手机被称作 Nexus One 的姐妹版，但是它加入了宏达电的 Sense UI 界面，因此支持多点触控。届时，所有购买了 Nexus One 的用户都会后悔自己出手太快，购买了一款所谓的 Android 旗舰产品。

但我们并不能通过宏达电 Droid Eris 和 Desire 这两款手机看出 Android 应用和 Android 系统本身的升级道路。由于配置了 Sense UI 界面，用户无法对 Android 操作系统进行升级，也无法确保所有的 Android 应用都能正常运行。毕竟，新的操作系统将与 Sense UI 界面冲突，而新的 Sense UI 界面又会与已安装的应用冲突。

这使得情况更加混乱。今年晚些时候，索尼爱立信将推出 Xperia Mini 系列手机，搭载 Android 1.6 操作系统，但使用索爱自己的用户界面。

这种情况给开发人员和用户造成了进退两难的局面，因为每一款新的 Android 设备都会增加新的变化，带来新的兼容性问题。与这种自我毁灭的行为相比，苹果的 iPhone 战略和 Palm 的 WebOS 战略显然更加明智：后两者均会在每年推出为数不多的设备，所有产品均搭载相同的操作系统，而且新款操作系统可以安装到以前的产品。因此，开发人员和用户都知道，自己所使用的手机平台非常连贯，是可以预见的。甚至连微软也认识到了这一点。在历经多年允许 Windows Mobile 平台自定义使用到大量相互不兼容的设备之后，微软即将推出的 Windows Phone 7 操作系统采用了与苹果和 Palm 相同的策略。同时，多年来一直同时采用多款黑莓操作系统的 RIM 也基本

上采纳了为所有设备配备相同操作系统的做法。

或许有人认为，苹果、Palm 和 RIM 比谷歌更容易保持平台战略的连贯性，因为它们都生产自己的设备。相比之下，Android 是在开源环境下研发的，每个设备生产商都可以自由选择自己需要的平台版本。这是一个政策问题，而非技术问题。微软在推出 Windows Phone 7 时已经总结了经验教训，认为自己即便不涉及硬件生产也可以强制执行连贯的、可预见的平台。再何况，这正是 PC 产业的一贯做法，并不是什么全新的理念。



几个月以前就有媒体认为，谷歌在开源 Android 平台的政策将给用户和开发人员带来极大困惑。当时有人建议，谷歌应当对 Android 平台进行管理，允许用户和开发人员购买设备和代码，以消除他们的担心。Nexus One 看起来好像是一次尝试，但实际上并不是。

这使得 Android 用户处于一种不安全的困境，有可能导致用户的离去。当然，谷歌想要学习桌面 Linux 开源系统的方法。然而桌面 Linux 用户至少不必绑定在某个版本的 Linux 系统上。在发现自己安装了无法更新的 Linux 系统之后，用户至少可以保住自己的硬件投资，只需要安装其它的操作系统。然而在智能手机领域，用户并没有这样重新选择的权利。

开源 CRM 软件商 SugarCRM 创下销售纪录

过去的 2009 年，对于开源 CRM 是跌宕起伏的一年。虽然开源 CRM 的功能不一定大而全，但对于普通企业来讲，它们所提供的功能已绝对够用。新的基于服务的软件应用模式，能让企业把资金只投资到他们所需要应用的功能服务上，无需再买回一大堆用不到的功能而浪费成本。这正是开源的灵活性与吸引力所在。

在 2009 年的 Dreamforce 年会上，SugarCRM 产品通过游击营销让我们见识到了其激进的表现。似乎这家开源 CRM 供应商在任何商业领域均极为激进，并在 2009 年创造了销售记录。

当大部分公司为了取得现金流，甚至为保持流动性而挣扎时，SugarCRM 在收入、客户群、社会及伙伴拓展等方面都迎来了创纪录的一年。去年一年，他们拓展了创纪录的 2000 名商业客户和 50 名新的增值业务中间商（VAR）；这些新的 VAR 也将 SugarCRM 的市场拓展到了超过 75 个国家。

Larry Augustin，作为去年一年间 Sugar 的掌舵者，对公司现有的表现极为兴奋，对于更出色的 2010 年，他们已做好充分的准备。2009 年，Sugar 与微软合作，提供了可运行于 Windows

Azure 系统的 Sugar 应用程序，同时他们也发布了可在 Amazon EC2 系统上运行的公众版。将公众版移植到 Amazon 的弹性计算云（Elastic Compute Cloud）上可使开发者连接、测试和开发 Sugar 代码，使其更容易获得。



如同任何其他出色的 CEO 一样，Augustin 表达了对即将于 2010 年发布的新产品的兴奋之情，而 Sugar 或将准备为 WOW 用户提供新的版本。根据 Jigsaw 最近的报道，SugarCRM 已成为全球前三大使用率最高的 CRM 产品，Sugar 或将成为公共配置的首选。

总体来说，开源 CRM 的出现使得 CRM 不再是大型企业独享的盛宴。

社区扫描

Fedora 对比 Ubuntu 六大要素决胜负

Fedora 和 Ubuntu 哪个更好？这个问题在 Linux 的用户中间问得越来越多。

这个问题不是偶然产生的，这两大发行版在过去两年里一直在下载排行榜上名列前两名。去看看 Fedora 和 Ubuntu 热火朝天的论坛，经常有绝望的用户咬牙切齿的宣布要转向另外一个。虽然其他发行版包括 Linux Mint、openSUSE 和 Mandriva 也提供了不相上下的功能，但只有 Fedora 和 Ubuntu 才被广泛视为桌面系统的最主要选择。

背景、目标和社区

Fedora 的前身 Fedora Core 是主要由 Red Hat 资助的。事实上，那些把全部时间放在 Fedora 上的人很多都是 Red Hat 的员工，而且 Fedora 的项目领袖和董事会全部 8 名成员中的 4 名都是由 Red Hat 任命的。

虽然 Fedora 的主要事务还是自己管理，但这种联系也遭致了一些批评，人们指责 Fedora 被当作是 REHL 的测试版，不过这也是事实。



同样，Ubuntu 和 Canonical 的关系也密不可分。Canonical 的创办人 Mark Shuttleworth 曾戏称自己是 Ubuntu 的终身监护人。决定大方向的 Ubuntu 技术委员会往往是由 Canonical 的雇员为主，而日常决策大多留给社区团队，这与 Fedora 没什么区别。

在过去五年里，Fedora 和 Ubuntu 都吸引了大量开发者，社区迅速扩展并分别拥有了自己的开发者大会：Fedora 的 FUDCon 和 Ubuntu 的 Ubuntu Developer Summit。两个社区的成员同时也是其他自由和开源软件大会特别是 GNOME GUADEC 的积极参与者。

总之，Fedora 和 Ubuntu 的结构已经惊人的相似。主要区别在于它们的目标：Ubuntu 的目标是成为“Windows 和 Office 的开源替代方案”，而 Fedora 的目标是“展示最新的自由和开源软件的 Linux 操作系统”。

安装和桌面选择

大多数用户都是用光盘来安装 Ubuntu 和 Fedora，一般都在半小时之内完成，包括一些安装后

的配置工作。为了以防万一，两个版本的安装程序都可以在文本模式下使用。Ubuntu 还配备了备用安装程序，可以更精细的按照每个部分来安装。

两个发行版还包括其他变种，包括 USB 安装和混合安装等。Ubuntu 还提供了用于 Windows 的 WUBI (Ubuntu Installer for Windows)，可以安装到一个现有的 Windows 分区，在开机时选择操作系统。

Fedora 和 Ubuntu 都把主要精力放在了 GNOME 桌面上，但也都准备了 KDE 和 Xfce4 包。事实上 Ubuntu 为其他桌面提供了衍生版也就是 Kubuntu 和 Xubuntu。这两个发行版特别是 Fedora 常被人认为过于专注于 GNOME 而忽视其他桌面。

附带软件

除了主题和桌面墙纸，Fedora 和 Ubuntu 的默认 GNOME 桌面仅在一些小地方不同。有几个工具安装在不同的位置，还有比如 Fedora 安装 Abiword 而不是 GIMP，而 Ubuntu 自带了 F-Spot 和 Xsane。但大部分的差异是如此的微小，你只需要 20 分钟的时间就可以让它们变得一模一样。

包管理和软件安装

在过去，Ubuntu 使用 Debian 软件包而 Fedora 使用 RPM 包，这是两个发行版之间的主要区别，但今天 Yum 的出现使得二者之间的区别几乎不再存在。

Fedora 和 Ubuntu 都采用图形化软件安装，这也使大多数用户在安装软件时不会注意到任何区别。因为 Ubuntu 是基于 Debian 的，它可能会有更多的软件供你选择。当然 Fedora 的软件选择也已经足够丰富。

SRS 实验室成为首家加入 LiMo Foundation 的音频公司

环绕声、音频及语音处理技术行业领军 SRS 实验室宣布它已加入 LiMo(Linux Mobile)Foundation™(LiMo 基金会)，成为首家跻身移动电话行业全球性协会行列的音频技术公司。

SRS 实验室市场部副总裁 Allen H. Gharapetian 表示：我们为得到“首家加入 LiMo 基金会的音频公司”这一殊荣而倍感自豪。我们口袋中的移动设备已极大地改变了我们取得与消费娱乐的方式。我们深信 SRS 的音频专长将把移动体验推至一个更高的创新境界。”

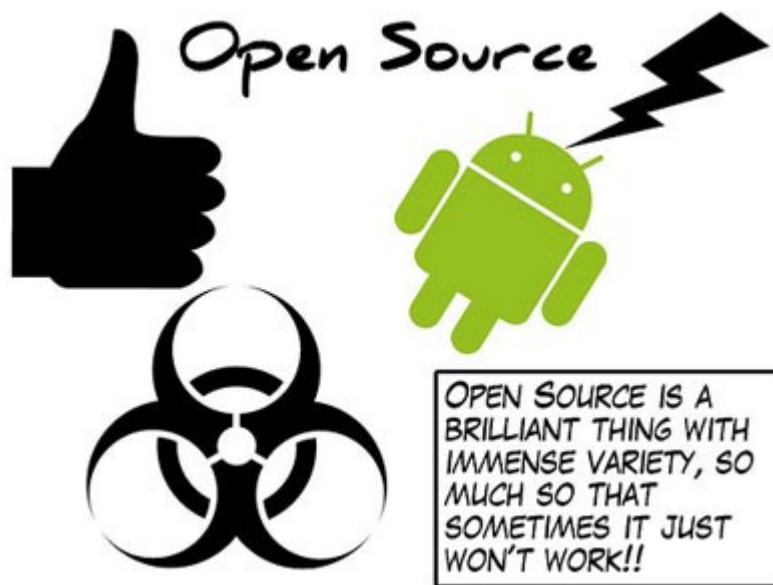
LiMo Foundation(LiMo 基金会)为移动电话行业提供了基于移动 Linux 系统的开放、稳定的手机操作系统，促成了下一代身临其境的移动消费体验的发展。SRS 现已通过在多款 NEC 移动电话中植入 VIP+™以及 TruMedia™，为 Linux 的移动操作系统应用作出贡献。

LiMo 的执行总监 Morgan Gillis 表示：“随着消费者对于深度临场式移动体验需求的增加，音频和语音技术现正成为移动家族中愈加重要的组成部分。SRS 的专业知识使得 LiMo 能进一步传递更丰富和互动性更强的体验。”

SRS 针对 LiMo 的音频解决方案的演示样本将于西班牙巴塞罗那 Mobile World Congress(世界移动通信大会)上展示，具体展示地点为 8 厅 B135 LiMo 展台内的 SRS 实验室分展台。同时，SRS 的会议室位于 7 楼 4 厅的 HS10 展台内。

Linux 社区为何驱逐 Google Android?

Linux 内核维护者 Greg Kroah-Hartman 曾宣布，从 Linux 2.6.33 版本开始，Google 智能手机操作系统 Android 核心代码将被删除。很多人对此十分关注，毕竟 Google Android 受到了很好的评价，作为 Linux 的一员能帮助开源社区实现更好的发展，对此，Greg 昨日在官方博客中作出了解释。



Greg 表示，自己使用的手机就是运行 Android 的 G1，而且它功能良好，自己非常满意。那么究竟是什么导致 Android 内核代码要被删除呢？“简单的说，没有人关心这些代码，所以它会被删除。正如我之前所说，staging tree 上的代码需要有人进行维护以便整合到 main kernel tree 上，否则将被删除。”

“Android 内核代码并不只是包括一些怪异的驱动，为了获得正常工作的 Android 系统，你需要获得 Google 创建的新的 lock type，以及其安全模式内核系统中的 hooks……Google 屏蔽了大量的硬件驱动，阻止平台代码整合进 main kernel tree。”

“我也不知道下一步会怎么样，Google 目前没有任何表示。很多企业现在正试图从他们的代码库中剥去 Android 接口，不过这也是一个非常大的工程，而且是不必要的痛苦过程。我真的很希望 Google 可以醒悟过来，去修复他们的代码库，实现它们的向上整合，消除他们现在对很多 Linux 硬件企业造成的壁垒。”

10 亿欧元才能开发出目前的 Linux 内核

根据西班牙 Oviedo 大学的研究人员的估计，Linux kernel 如果在欧盟内开发的话，它的开发成本预计将超过 10 亿欧元（约 92.4 亿人民币），需要开发者 986 人，用 14 年时间去开发。

研究人员估算的 Linux kernel 年度研发费用分别为：2005 年 8 千万欧元，2006 年 7300 万欧元，2007 年 9500 万欧元，2008 年 2.28 亿欧元。以 2008 年为例，其研发费用相当于微软产品研发费用的 4%，Google 的 12%。开发者的平均薪水为 31040 欧元。在此之前，也有一些研究人员估算了 Linux 发行版 Red Hat、Debian，和 Fedora 的研发费用。

发行版发布：PC-BSD 8.0

针对桌面用户设计的 FreeBSD 变种 PC-BSD 现已发布 8.0 (Hubble Edition) 版本。PC-BSD 8.0 基于 FreeBSD 8.0-RELEASE-P2 和 KDE 4.3.5, 包含许多增强和改进。

PC-BSD 8.0 的显著变化包括：

- 全新的系统安装器，允许安装 PC-BSD 或 FreeBSD
- 直接从 DVD 运行 Live 模式
- 对软件管理器进行了更新，可直接浏览和安装应用程序
- 在 amd64 平台上支持使用 NVIDIA 驱动的 3D 加速特性

参阅其[发布公告](#)和[发行注记](#)可了解详情。你可以从这里[下载 PC-BSD 8.0](#) 的 ISO 映像。

你在使用开源软件就是用盗版？

一个美国版权游说组织试图让美国政府认识到开源等同于盗版。

由出版商、软件、电影、电视和音乐协会组成的国际知识产权联盟 (IIPA) 向美国贸易代表办公室 (USTR) 提出要求，考虑让印度尼西亚、巴西、印度等国列入《特别 301 报告》的观察名单，因为这些国家鼓励使用开源软件。

《特别 301 报告》是关于世界各国的知识产权保护，主要针对的是盗版软件、盗版光盘等问题。去年美国发布在《特别 301 报告》中，中国在优先观察名单中排在第一位。在 IIPA 的努力下，去年加拿大也被列入了“优先观察名单”。

Novell 和 LPI 开展培训和认证合作

近日，Novell 公司和 LPI 于近日宣布，双方将开展国际合作，对各自的 LPIC-1 入门级 Linux 认证项目进行统一。根据该项目，已经获得 LPIC-1 身份的 Linux 专业人员也将同时获得 Novell® 认证 Linux 管理员 (CLA)。另外，Novell 培训服务也已正式同意在其 CLA 课程培训材料中加入 LPIC-1 学习目标。

“作为一家经过 25 年考验的企业软件和服务提供商，以及最具互操作性的 Linux 平台提供商，Novell 具有良好条件提供综合全面的培训项目，可确保 Linux 专业人员能够实现全球认可的 LPIC-1 认证，使他们能够身怀专业技能和信心，应对多种多样的数据中心要求。” Novell 开放平台解决方案总经理兼高级副总裁 Markus Rex 说。

业界需要具有最大可靠性、可管理性并节省费用的解决方案，随着这一需求的不断增加，包括 Novell 的 SUSE® Linux Enterprise 在内的 Linux 的采用步伐正在加速。一项 IT 管理人员 2009 全球调查显示，40% 的参与者计划在未来的 12 至 24 个月内，在 Linux 上部署更多的工作负荷，而 49% 的参与者表示，在 5 年之内 Linux 将是他们主要的服务器平台。

“此次合作表明 LPI 认证项目在更广阔的 IT 和 Linux 社区内拥有强大的支持。这种历史性的支持包括来自诸如 Novell 这样的厂商的贡献，并已促成 LPI 成为最广泛认可和接受的 Linux 认证项目。” LPI 总裁兼 CEO Jim Lacey 说。“我们期望与 Novell 合作，共同促进 Linux 未来力量的进一步发展。特别要提到的是，通过在培训和考试准备课程中支持 LPIC-1 目标，Novell 已经承认了业界对于

与厂商无关的认证项目的需要，这些项目将帮助 IT 专业人员在企业环境下可自如的应用任何 Linux 发行版。”



根据该合作协议条款，所有 LPIC-1 资格拥有者都将有机会无需额外考试或费用而申请 Novell CLA 认证。Novell 培训服务将把 LPIC-1 目标加入到其 Linux 管理员课程和项目中，其中包括自学、按需和由合作伙伴引导的课堂培训。

“作为一家经过授权认证的 Novell PartnerNet®项目培训合作伙伴，SwitchDesk 已经对来自国内顶级公司的一些 IT 人员进行了培训，诸如福特汽车公司、波音公司、Macromedia、NASA 等许多其它公司。” SwitchDesk 公司总裁兼 CEO Chris Clark 说。“我们很高兴看到 LPI 和 Novell 合作开发统一协调的课程内容目标，并提供双重认证，这样我们的客户将具有广阔的技能覆盖面。”

Apache Http Server 的 15 岁生日

近日，Apache 基金会的网站刊登了 Sally Khudairi（Apache 基金会市场与公众事务部经理）的一封邮件：Apache 基金会宣布 Apache HTTP Server 已 15 周年，以庆祝 15 岁生日。



2009 年末，Apache 基金会刚刚度过其 10 周年。其实早在 Apache 基金会成立前，Apache HTTP Server 已经存在很久。如今这个世界上最著名的 Web 服务器在为超过 1 亿 1200 万的站点服务着。

"1994 年 2 月 23 日，世界各地的个人"补丁"作者受邀参加 "new-httpd" 的邮件列表，来讨论 NCSA httpd 的未来版本和性能加强。经过几天的讨论，Apache 的名字被选择做为新的努力结果，用标准、开源、可拓展的软件系统代替已有 NCSA 服务器。..."

Sally Khudairi 在邮件中生动描述了 Apache 的诞生、历史、成就，查看邮件内容。

自由软件基金会：Google 应该开源 VP8 编码器

全球自由软件的主要推动者自由软件基金会(FSF)近期在网站发表了一封致 Google 的公开信，敦促 Google 开放刚刚收购的 On2 公司的高性能视频编解码器 VP8 的源代码，并在 YouTube 上使用，结束互联网对专利视频格式如 H.264 和专有软件 Adobe Flash 的依赖，开创真正自由的网络。

自由软件基金会在公开信中称，这两个举措对 Google 来说，都易如反掌，但是小行动却可以带来巨大的影响。



通过在 YouTube 上支持新的自由视频格式，并抛弃 Flash，全面转向 HTML，将吸引大量设备厂商和浏览器等软件应用转向自由技术，从而使其成为真正的标准，促使更多软件开发者和互联网用户转向自由技术。这样，整个社区都将彻底摆脱专利的困扰。而 Google 也将巩固自己的统治地位。

目前，Google 尚未对此做出回应。Google 本周刚刚完成对 On2 的收购交易，金额达到近 1.25 亿美元。On2 公司是一家专业视频压缩技术公司，客户包括 Skype，Nokia，Sony 和 Adobe。

Symbian 操作系统提前完全开源计划

据国外媒体报道，Symbian 基金会近日提出推进 Symbian 智能手机平台完全开源，比预计的发布日期提前 4 个月。

Symbian 3 平台，包括应用软件，中间件以及内核本身，将提供 Eclipse 下的公共许可和其他开放源码许可证条款。Symbian 基金会的全球联盟负责人 Larry Berkin 先生表示：“你可以下载它，你可以修改它”。在此之前，内核在开源也是可以获取的。

Berkin 先生表示：“在源码级别，我们开放 108 个包，手机制造商可以修改代码，建立差异化

的手机。原定于 6 月全面开放，然而基金会的成员加速这一进程。在太平洋时间早上 6 时，有超过 4000 万行的代码，将在 Symbian 的网站可以获得。终端用户将看到，理想情况下，不同特色的设备，多功能设备将会是基于 Symbian 平台，范围从智能手机到多功能设备，如数码相机或手机，游戏机设备等，开源可能会导致不兼容以及平台版本的分支，制造商需要与 Symbian 一起对他们的工作负责。像这样的情况，市场能够自我纠正。”

Berkin 先生说：“目前有 33 亿部基于 Symbian 的设备在使用。目前，有五家制造商制造的设备使用 Symbian 平台：诺基亚，三星，索尼爱立信，富士通和夏普等。”

Symbian 有限公司推动了 Symbian 技术的发展，其中诺基亚占据主要地位，然后作为一个开源项目出来推出。



IDC 公司的移动设备技术和趋势的项目经理、分析师 William Stofega 称：“将 Symbian 平台推向开源将会提升 Symbian 平台的市场地位。我认为这不仅对诺基亚是好事，它也对 Symbian 非常的有益，这会提高 Symbian 平台所有在其可行性方面的市场份额，提高和 iPhone 以及 Android 的竞争力。”

同时可供下载的是建立 Symbian 的应用程序和移动设备的开发工具包。这些包括 Symbian 开发工具包(SDK)和产品开发工具包(PDK)。

2009 年 11 月，诺基亚公司推出的高端 N900 “移动电脑”，他基于 Linux 的 Meamo OS，其特点是具备电话、电子邮件、Web 浏览器和视频功能。但该公司依然是 Symbian 的支持者。

行业观察

Google 店大欺客 伪开源 Android 危机四伏

据媒体日前报道：Google 公司日前单方面取消了中国联通以及两家手机制造厂家联合开发 3G 谷歌手机(Gphone)的计划，并从上周起中断了每周三举行的“四方沟通会”。

Google 公司不再授权使用谷歌标识及应用产品的合作内容。原定于 1 月 20 日举行的 WCDMA 谷歌手机(Google Phone)上市发布会也在谷歌建议下取消。参加该项目的手机制造企业摩托罗拉公司，也于同日接获 Google 通知，要求将为中国联通合作项目设计、并已经批量生产的手机产品返厂处理，“去除所有 Google 因素”。据称另一家参与合作的三星公司也接到了相同的通知。

Google 并未在对上述两家手机厂商自由使用 Google 手机平台 Android 提出任何限制，但明确要求，中国联通谷歌项目推出的手机产品在外观和品牌名称设计上，将不能再使用 Google 标识、英文 Google Phone 以及中文“谷歌体验手机”的称谓。手机软件系统中已经预置的搜索引擎、地图等大量谷歌产品应用也要全部删除。

事情的性质

Android 手机平台是一个 Google 的开源项目，而就开源项目来说，不仅仅是意味着它们源代码是公开的，可以任由用户使用和修改，并且还意味着，所有的用户都有权使用它，也就是说，开源项目的拥有者并无权对用户进行干涉，也无权对用户进行挑选，不能够说，我看张三顺眼就让张三用，我看着李四不顺眼就不让李四用。



而且，对于开源项目所集成的应用，除已有特别声明的之外，一律开放使用权限。当然，作为 Android 手机平台的拥有者，Google 是有权对用户实施一些自己认为需要的歧视性限制，但这是另外一个概念，说明 Android 手机平台已经不属于真正的开源项目了。

而这次问题的关键是：Google 要求摩托罗拉和三星将为中国联通合作项目设计、并已经批量生产的手机产品返厂处理，“去除所有 Google 因素”。手机软件系统中已经预置的搜索引擎、地图等大量 Google 产品应用也要全部删除。的确，从拥有者的角度来说，Google 是有权这样做，但是，从一个开源项目来说，Google 是不应该这样做。

甲骨文收购 Sun 消化不良 Sun 赢利困难

近日，Oracle 收购 Sun 得到欧盟批准。埃里森在回答媒体提问时表示：“我认为目前的合作进展很顺利，预计 Sun 将很快实现盈利”，不过分析师认为，这只能再次说明埃里森是个很有野心也很喜欢夸口的人。

因为在此之前，甲骨文也曾在另外的并购案中这么宣称过，然后事与愿违。何况 Sun 的情况并不乐观，之前甲骨文的高管也一直拒绝对 Sun 实现盈利的时间做出预测。

分析师认为，埃里森很难在短期内使 Sun 扭亏为盈，之前甲骨文承诺 Sun 在被收购一年内将带来 15 亿美元的运营性利润。但在被收购的前两年中，Sun 出现了超过 20 亿美元的亏损，自本世纪初互联网泡沫破灭以来，Sun 一直未能实现复苏，除了硬件部门，Sun 本身没有盈利的能力。即使是 Sun 引以为傲的 Java 和开源，除了能够给 Sun 带来无限的口碑之外，一直没有寻找到自己的盈利模式。



甲骨文并不善于低端作战。Sun 同样不善长在低端市场与竞争对手火拼。甲骨文和 Sun 合并之后，Sun 定位于低端市场、“薄利多销”的产品都将有可能面临被弱化的命运。

Sun 的硬件部门更是让甲骨文头疼，甲骨文在此前并未有硬件业务的经验，虽然甲骨文高管在电话会议上表示，可以向包括电信、零售和银行业在内的行业捆绑销售 Sun 的硬件与 Oracle 软件。但 Sun 在全球服务器市场的份额只有区区 4%，其大部分营收来自于与 Sparc 芯片相关的服务器业务。实际上，由于基于英特尔和 AMD 芯片的行业标准服务器常常更为便宜，消费者已纷纷放弃 Sparc 服务器。埃里森在接受采访时曾说不会退出硬件业务，如此一来，继续维持服务器业务势必伤害到甲骨文高达 46% 的营业毛利率。

由此，一些华尔街分析师和业界分析师甚至建议甲骨文出售 Sun 硬件业务，集中精力搞有长期发展前景的软件业务。

然而，在软件层面，MySQL 被称为是“最受欢迎的开源数据库”，并购后也将面临被弱化的命运。开放源代码软件社区领袖 Richard Stallman 在给欧盟委员会的信中表示，“如果 Oracle 是今日旧数据库市场的统治者，MySQL 就是未来新兴数据库市场的主导者，MySQL 可以被视为 Oracle 未来最重要的竞争对手。”一旦 MySQL 被 Oracle 所合并，Oracle 不仅没有义务积极推销

MySQL，或以合理价格提供其商业授权，更没义务拿自己的营收来改进 MySQL。面对商业数据库软件与开源数据库之间的利益冲突，恐怕 Oracle 会优先保护其收费的数据库产品。

曾是 MySQL 股东和顾问的 Florian Mueller 表示，“让 Oracle 得到 MySQL 比让狐狸负责鸡舍更可怕，因为母鸡没有对狐狸构成威胁，而 MySQL 却让 Oracle 丢失客户，并不得不给一些因威胁可能离开的客户以折扣。”这与 MySQL 创始人维德纽斯一直不支持收购到处发起签名拯救 MySQL 一样，他也是因为担心并购后 MySQL 面临被消灭的命运。

一位不愿意透露姓名的甲骨文相关人士透露：收购后甲骨文有可能对自己原有的数据库和收编进来的 MySQL 进行重新定位。甲骨文数据库锁定高端市场，MySQL 定位在中小企业市场，但 MySQL 也将面临与甲骨文数据库中小企业版的冲突问题。

去年 10 月 27 日，Red Hat 宣布以 1900 万美元注资开源数据库厂商—EnterpriseDB。消息一出，立刻被业界解读为 Red Hat 另辟蹊径发展开源数据库，不想以后受制于 Oracle 和 MySQL。

这样，Java 将成为埃里森的一张王牌，分析师认为甲骨文掌控了 Java 就等于掌控了将来的产品维护趋势，将增加用户的忠诚度。可还是那句话：“当开源的 Java 遇到商业的甲骨文，谁都无法预料最后的结果。”而 Solaris 操作系统和 Sparc 服务器曾经是甲骨文数据库的主要平台，这对甲骨文极具诱惑力，埃里森谈到 Sun 的 Solaris 时，也说它是迄今为止最好的 Unix 技术，意思似乎是想依托 Solaris 大有所为。但事实上，在全球三大操作系统市场，来自 Unix 的市场份额正在不断遭到 Windows 和 Linux 的蚕食。而且，Oracle 在内部使用的是 Linux，甚至有自己的 RHEL(Red Hat 企业版 Linux)和坚不可摧的 Linux 计划。

Symbian 开放源代码 开源是否等于成功？

2010 年 2 月上旬，Nokia 旗下 Symbian 协会宣布，该协会已经完成了 Symbian 平台的源代码开源发布。这一目前智能手机中应用最广的平台目前已经实现完全开放，平台源代码可免费使用。



平台开源将成为 Symbian 发展史上最大的一次改变，这一进程比此前计划的要快。诺基亚在 2008 年全面收购 Symbian 之后，表示要将 Symbian 操作系统打造为一个开源系统，并将其源代码公布于众。按此前的计划，诺基亚会在今年六月完成这一进程，如今，Symbian 协会提前四个月完成了这一进程。

开放源代码之后，所有的个人和组织在未来都可以开发基于 Symbian 操作系统的产品，并且无需任何授权费用。这意味着，新的 Symbian 操作系统将和谷歌的 Android 操作系统形成直接的竞争，后者的开源操作系统目前正发展势头强劲。

Symbian 是目前用户数最大的智能手机操作系统，其产品总数量达到 3.3 亿部。然而，在近两

年来，Symbian 平台的市场份额已经被 RIM、苹果以及 Android 等蚕食。此次开放源代码，将是 Symbian 平台崛起的重要一步，但是，开源真的就意味着成功吗？

开源软件不会比“闭源”软件安全？

网友：morgancheng

自由软件的斗士 Eric Raymond 把这句“足够多的眼睛可以让所有的 bug 无处遁形”称为“Linus 定律”，也是证明开源软件比“闭源”软件更有优势的一个论据。最近微软一位安全方面的资深工程师 Shawn Hernan 在博客中批驳了这一论断，作为“闭源”软件的代表，微软工程师这么说不奇怪，但是 Shawn 的写作水平很高，这篇博客可以作为“议论文”的范文。

要驳倒对方观点，秘诀不是去正面和对方交锋，而是绕开其锋芒，去挑战其前提假设，Challenge the assumption。看样子 Shawn 也是深谙此道，直接找到了“Linus 定律”的 Assumption：

1. 开源软件比“闭源”软件有更多的 Code Review；
2. Code Review 可以让软件更加安全。

逻辑学上的三段论，结论就是“开源软件比闭源软件更加安全”，Shawn 指出这两个前提假设 (assumption) 都是不成立的，所以这个结论也是不正确的。

干得好，Shawn！真希望每一个程序员都能有你一样的 Critical Thinking 和论证能力，所以，从我做起，我们也来 Challenge 一下 Shawn 的 Assumption：)

首先，人家“Linus 定律”不是专门针对安全性 bug 的，你为什么假设人家说的是安全性问题呢？这世界上除了安全性 bug 还有很多其他的 bug 呢，你是不是评理找错了对象呢：)

Shawn 的潜台词是安全性问题高于一切，其实开源的目的并不是要造出最安全的软件，而是通过共享源代码鼓励创新和快速的软件开发（你也可以 Challenge 我这个 Assumption，但至少我现在是这么理解开源的）。

回到安全性问题本身，我个人觉得在安全性上开源软件比“闭源”软件没什么必然的优劣之分，安全性和其他任何软件的特性一样，决定于项目的重点和倾向。源代码是程序员写的，程序员会根据工作重点倾向于加强某些特性。如果某个软件项目强调代码可读性，那么可读性就会强一些；如果某个软件项目强调运行效率，那么运行效率就好一些；如果某个软件项目强调安全性，那么安全性自然也会好一些。

从这点上开，我相信微软的软件安全性会好一些，因为毕竟微软很强调安全，而且也不是光说不练，花了不少力气整治安全性。我知道我知道，微软产品不断爆出安全漏洞，也不断地在打补丁，我只能说：一切创造过这么庞大软件项目而没有造出一个安全漏洞的同志，都有资格嘲笑微软。

本期推荐

开源：“移动航母”的核动力

如果把移动行业比作迎着朝阳的航空母舰，那开源就是驱动航母快速前行的燃料核动力。开源与移动发展似乎是天造地设的一对，无论如何，开源都是移动更快发展的关键。从 Gartner 预测的 2012 年数据来看，考虑发展、转型等因素，开源智能手机操作系统所占市场份额将高达约 60% 以上。



开源在移动行业中不断壮大。例如 Android，从其出生伊始就受到热捧，并日益普及。LiMo 基金会和 Symbian 的完全开源，开源被证明无处不在。开源平台帮助摩托罗拉走出困境，其 Droid 手机上季度季报表现不俗，扭转了公司前景。开源平台还受到正陷于亏损的索爱的青睐。即使闭源 iPhone 其多个组件也是基于开源的。三星也推出了自主开源手机系统操作系统 bada。

移动行业也许比任何其他行业更适合开源，是因为它的创新合作发展模式。开源的 Symbian 操作系统的手机市场份额 46.9%；Android 手机则是去年同比增幅最大的，其市场占有率提高了 7 倍，达 3.9%。分析师预计 Android 在 2010 年将获得更多的市场份额，

智能手机在重大社会活动中扮演着越来越重要的角色，从海地地震到美国大选都有一些共同的特点那就是智能手机。全球各地数十亿的人专注地盯着自己的手机，通过手机账户捐赠善款给慈善机构和更新 Twitter 的有关海地地震的信息，参与美国大选互动。根据美国非赢利性组织“手机给予基金会”报告称，截至 1 月 15 日，美国手机用户以发送短信方式向海地灾区捐款超过 1100 万美元。而手机也在美国大选中争得一席之地。有报道称当传统媒体记者等待奥巴马宣布他的竞选搭档时，奥巴马的回答是“等我的手机短信。

当然，并非每个人都有手机，但手机的日益普及，是不容忽视的。许多分析公司，包括 Gartner 和 IDC 的预测，智能手机不仅将继续受到消费者欢迎，而且未来也将超过 PC 销售，尽管目前世界正处在经济危机的浪潮中。

为了满足预计的市场需求，智能手机制造商不断推出新的机型。因此，设计，生产，应用开发，

都必须跟上市场需求的节奏。

开放的智能手机操作系统平台，将吸引第三方开发者、手机制造商和用户强烈关注、访问并登录开发、修改应用程序，可以让开发者很方便地测试自己的现实增强软件，这样有利于壮大开发力量，发展生态系统和扩大用户群。

开放源代码的使用将是极大的提高软件开发效率的关键。在现如今的软件开发环境中，很少有程序员是完全从头开始一个项目。Gartner 的调查显示，49.7%的开源软件应用是用于重要任务的应用程序。相比之下，专有软件有 59%用于重要任务应用程序，有 58.8%用于内部开发。

现有开源组件，方法，类和算法是不胜枚举。调查显示开源程序及项目多达 22 万种，可以通过超过 4500 个网站进行下载，这些开源软件组件，能够有效降低软件开发成本，消除创新浪费，帮助企业延伸发展资源，但更重要的是，开源提高开发的灵活性和敏捷性。

在迎接移动互联网的大背景下，尽管开源已经被广泛采用于移动通信行业，但这并不意味着项目将始终顺利运行。诸如管理，缺乏标准化，法律法规和安全等关键因素如果得不到妥善解决，那么开源这个核动力也是会变成核炸弹。

英特尔的 MeeGo+X86 能否撼动 Android+ARM?

目前的手持市场，Android 在操作系统火透半边天，ARM 则在硬件平台火透半边天，那么 Android+ARM 会不会一手遮天？不会，因为有很多人不愿意，并且还是举足轻重的人。这不，英特尔与诺基亚携手搞出一个 MeeGo 操作系统，也是基于开源 Linux，也是要在上面跑应用商店（App Store），明眼人一看就知道，这个新的操作系统就是来对抗 Android+ARM 的。当然英特尔与诺基亚在此次合作中各有心事，前者是要反攻 ARM 阵营，而后者则是要借 MeeGo 来发展高端智能手机市场，由于 Symbian 不够开放，上面不能跑开放的应用商店，诺基亚更不可能用 Android，所以借助 meeGo，可以让诺基亚向开放的应用商店模式发展，也圆了诺基亚一直想走的服务之路，将来有可能与宿敌苹果一拼高低。

“MeeGo 不会取代 Symbian，它会与 Symbian 互补。”英特尔中国区总裁杨叙在日前接受专访时谈到。那么英特尔从此次合作中得到什么呢？

MeeGo 是由英特尔的 Moblin 和诺基亚的 Meamo 两大软件平台合并而来，它采用了 Moblin 的核心操作系统，同时借用了后者的 Qt 应用开发环境。所以，这里最重要的是英特尔将 Qt 这个在软件领域广泛采用的应用开发环境引入到 X86+MeeGo 平台上，并且也将诺基亚的用户界面框架引入 X86 平台；此外，借用与诺基亚的合作，将之前在小范围内采用的 Moblin 操作系统扩展到众多电子设备领域，从而可以为英特尔走向服务之路打开大门。“将来基于 Qt 开发的应用程序可以同时跑在 MeeGo 和 Symbian 等多种平台上。”杨叙说道，“更重要的是，MeeGo 上开发的 App Store 不仅可以用于手机，也可以用于互联网电视、机顶盒、车载娱乐等多种 X86 设备上。”他的一番话道出了英特尔向服务领域迈进的重要策略。

苹果通过服务取胜的商业模式成为所有企业模仿的对象，英特尔也不例外。“其实，英特尔很早就倡议增值服务，但是 iPhone 之前，没有任何成功案例，而且没有任何人相信增值业务是一个很好的东西，好象可望而不可及，好象在追彩虹一样，永远追不着的东西。现在，苹果的成功，为大家指明了一个方向。Google 也要做了，大家都要做了。”杨叙说道，“不过，苹果的东西不开放，厂商与他合作很难，所以现在大家都选择 Android。”

那么，与 Android 相比，MeeGo 的优势在哪里呢？“MeeGo 比 Android 更加开放，

Android 的 App Store 必须基于 Android 系统开发，并且只能跑在 Android 系统上，这对软件开发者而言是一个门槛，对用户则是一个制约。而 MeeGo 将会更开放，它可适合于 X86 和任何其它架构的处理器，MeeGo 开发的 App Store 将可适合更多设备，比如为手机开发的 App Store，移植到互联网电视机上也可以采用。”他解释。所在，MeeGo 的推出将创建一个非常繁荣的 App Store 服务产业，这会给中国厂商带来非常大的机会。

Linux 操作系统新贵 MeeGo 的前世与今生

在刚刚结束的巴塞罗那 2010 世界移动大会上，Linux 智能操作系统家族两股势力合并，即英特尔的 Moblin 与诺基亚的 Maemo，合并后启用新名 MeeGo，并交由 Linux 基金会管理。

MeeGo 问世新闻一出就如猛虎下山，引发业界广泛关注和热议。谁说英雄不问出处？先不论未来是否三分天下而得其一，正是因为其出身显贵，MeeGo 才吸引了众多技术人员、分析人士以及媒体记者，在从来就不缺乏竞争和热闹的移动计算领域，受到格外眷顾。

很显然，MeeGo 是肩负使命和承载冀望而来。从产业角度的解释，其使命是促使家庭、车载、办公室和便携式等多种设备之间进行无缝通信，广泛应用于便携式笔记本电脑、上网本、平板电脑、多媒体电话、联网电视机和车载信息娱乐系统等；从两家合作者的角度，无疑是英特尔和诺基亚想在 iPhone、Android 风光无限之时夺得自己的一片领地。不仅英特尔和诺基亚，有如此志愿的还有微软，寄望 Windows Phone 7 打个翻身仗。

然而，MeeGo 背后有怎样的故事？MeeGo 究竟为何物？其因何而来，又向谁而去？本文将与您一道探讨。

Linux OS 新贵 MeeGo 问世

MeeGo 出生原因，说来简单，英特尔和诺基亚要合力专注于一个共同的平台，携手谋求更大的事业，而不是单独煞费心神地烹调两份的羹汤。



Forrester 分析师 Ian Fogg 指出，两者确实有很多共同点：同样是基于 Linux；同样都是主攻移动设备；同样的致力于提供杰出丰富的消费者互联网体验。此外，IT 界有部分观点认为，两家的 Linux 系统 Moblin 和 Maemo 之前发展速度都让巨头们有些失望，缺乏手机厂商的支持，诺基亚和英特尔急需转身求变。

连诺基亚自己都承认，从 Maemo 第一个版本面世到搭载到第一款智能手机 N900，时隔有五年。英特尔的 Moblin 可以让开发人员快速开发相关设备驱动及 UI，其 Moblin 社区也一直很活跃，

但是基于 Linux 的内核似乎没有什么特别优势，遗憾是，目前还没有智能手机硬件厂商支持 Moblin，这对于在计算设备领域前呼后拥的英特尔而言，显然是亟待解决的问题。

时间到了 2009 年 6 月，两家联手共书大计昭示天下，声明战略合作关系。双方宣布结成长期合作伙伴，研发下一代基于英特尔构架的无线计算设备和芯片组构架，以及 Linux 项目的合作，英特尔还将获得在未来产品中使用诺基亚 HSPA/3G 调制解调器的许可。

融合后的 MeeGo 问世，应该算是行动上迈出的重要一步。

从 Moblin 和 Maemo 看 MeeGo

MeeGo 继承了分别掌握底层架构和终端设备优势的两大巨头的优秀基因，该操作系统整合了两家各自的 Linux 运算环境（英特尔 Moblin 与诺基亚 Maemo），针对多种计算设备的硬件平台而设计，包括便携式笔记本电脑、上网本、平板电脑、多媒体电话、联网电视机和车载信息娱乐系统等，志在挑战抢尽风头的 Google、Apple 等终端平台及操作系统。

据称 MeeGo 融合了 Maemo 和 Moblin 两者优势，是面向多种处理器架构的开放式平台，MeeGo 保留了 Moblin 核心操作系统的能力，支持广泛的设备类别和丰富用户体验，并结合了 Maemo 在移动产业的影响力，以及已被软件开发人员广泛采用的 Qt 应用和用户界面框架。开发人员现在就可以用 Qt 编写 MeeGo 应用程序，而 MeeGo 的首个版本将在 2010 年第二季度发布。

针对部分厂商担心的诺基亚和英特尔将通过 MeeGo 在未来移动互联网时代获得的控制力和影响力，英特尔曾强调，MeeGo 非英特尔和诺基亚独有，而是定位在开源软件，敞开大门广迎宾客。原 Maemo 社区委员会主席 Valério D Valério 在谈到这个问题时也表示称，“鉴于 Linux 基金会将领导这次合并，而且他们过去的表现已经证明了自己，我想这问题应该也不太需要担心”。

从 MeeGo 的架构图和相关资料来看，MeeGo 分为三层：由 Moblin 的内核为主的 OS 基础层，MeeGo UI 框架和其他服务构成的中间层，以及针对不同设备的用户体验参考设计。

MeeGo 的应用方面会有飞跃，原有的 Moblin 和 Maemo 应用都可以直接运行在 MeeGo 上，近千个基于 MeeGo Qt，WebKit 的开发应用，多个 Runtime 之上的应用等。同时业内分析人士称，从硬件芯片的角度来看，合并后的 MeeGo 不仅支持基于英特尔架构的芯片（这是在笔记本、上网本、MID 等领域得到广泛应用的处理器），也支持 ARM 芯片，而 ARM 芯片广泛应用于智能手机中。这将为 MeeGo 增加一份独特吸引力。

再看看搭载 MeeGo 操作系统的终端设备开发进度。诺基亚、英特尔两家宣称 MeeGo 将由 Linux 基金会管理，遵循开源开发模式的最佳实践，首个版本将在 2010 年第二季度发布，而 MeeGo 设备将在 2010 年下半年亮相。计划终归是计划，合并后的 MeeGo 在终端的进展如何，还要看这两个玩家的决心和努力程度，让我们拭目以待。

MeeGo 引发智能操作系统新竞争

这个强强联手的故事，貌似也是一个无奈之举，iPhone 大概是其直接促成者。

在北美市场，让诺基亚遭受重挫的，那就是 iPhone。Forrester 分析师 Ian Fogg 在博客中提到，这一合并是“勇敢的玩法”，将使 MeeGo 赢得与 Android、iPhone OS、Chrome、Palm Web OS、Windows Phone 甚至 Ubuntu 桌面平台竞争的优势。但他同时也认为，诺基亚需要一个快速调整，从“说”到“做”甚至跑起来，否则欧洲高端市场也将如北美市场一样成为其他手机厂商的领地。

在智能终端的操作系统竞争中，除了 iPhone OS，当然还有风头正劲的 Android，以及因为大

幅改进的新版本而正在恢复竞争力的 Windows Phone。微软与英特尔、诺基亚在 OS 的新姿态，将为其赢得新的竞争优势。J.Gold Associates 分析师 Jack Gold 撰文称，“对于微软、英特尔和诺基亚来说，这确实是一个重新开始，不过微软在更为不利的一个位置，要从更陡的道路上爬回竞争轨道。”

Android 有其优势：合作伙伴多、品牌效应好，是支持 ARM 芯片架构的 Linux 操作系统，平台公布一年后即取得了智能操作系统 3% 的市场份额，甚至媒体更积极的报道称是 7%。但 Android 也不是无可挑剔，例如，缺乏像苹果、诺基亚那样的品牌影响力，使用者抱怨操作界面和方式都不够友好。

另外，从开发人员的角度，“Android 系统局限很大，开发者无法进入被 Google 牢牢控制的底层，这有点吓人。” Maemo 社区委员会主席 Valério 在比较 Maemo 和 Android 时有如此陈述。在他看来，Maemo（以及 Moblin 和 MeeGo）基于 Linux，有无限的可能性，使用的自由度更大，可以支持整个设备，而 Android 的开放还不够透明。

MeeGo 平台改写未来格局？

无疑，MeeGo 平台将成为一股新势力，加入到智能终端 OS 的激烈竞争中，其未来发展留有极大想象空间。改写智能操作系统格局？一切皆有可能。

分析师 Ian Fogg 给出建议，MeeGo 不应该一下子把注意力放在过于广泛的设备上，如车载信息娱乐、智能手机、上网本、Tablets 等等，而应该集中在一个领域如移动市场取得快速进展。

最新消息对 MeeGo 是乐观的。就在 MeeGo 问世两周多后，法国电信运营商 Orange 便与英特尔签署了一项战略合作协议，双方携手计划推出配置英特尔 Atom 处理器并运行新的 MeeGo 平台的移动电话和上网本，两家公司还表示，会在 MeeGo 设备上为用户提供类似 Orange 地图和 Orange 电视等多媒体服务。

Orange 是第一个正式宣布支持 MeeGo 的电信运营商，紧接着又引来第二家运营商关注，即意大利电信。意大利电信与英特尔将合作开发运行 MeeGo 系统的下一代互联网和网络电视服务，这一合作的目标是通过结合网络 TV、视频点播以及基于网络的更多服务，使多媒体内容的发送和接受更方便、更有吸引力。两家公司还透露，第一款运行 MeeGo 的产品名称为 CuboVision，采用 Atom CE4150 处理器，这款新的多媒体宽带设备将通过数字地面频道带给消费者免费和收费的电视内容。CuboVision 不仅提供来自数字地面电视、网络电视、视频点播及在线游戏等娱乐内容，还支持电子政务、电子医疗、电视商务、家庭自动化、远程教育和远程视频监控等应用。

不过，长期发展关键还要看执行，评论者的担心就在于英特尔和诺基亚的行动能否跟得上计划，足够快速且卓有成效。分析人士称，如果英特尔和诺基亚打算认真建立一个移动平台，应该可以办到，未来发展事实将会证实一切。

技术新知

Linux 大史记---缅怀已逝的十八年

fudan_abc 之任桥伟

至此落笔之际，恰至 Linux 问世 18 周年，18 年的成长，如梦似幻，风雨颇多，感慨颇多。

犹自忆起多年以前一位前辈训导时的箴言：今天的必然正是由之前一系列的偶然所决定的。过去的某年某月，我偶然初识 Linux 就身陷其中，至今仍找不到出去的路，而正是这次乃至之后的多次偶然相联合，从而决定了今日的我要在此写下这些话。那么，当您偶然地拿起这本书，偶然地看到这段话，您是否会问自己：这样的偶然又会导致什么样的必然？

如果您依然决定继续这次的偶然之旅，那么首先请认识两个人，准确的说是一个人和一只企鹅。这个人自然就是 Linus Torvalds，我们也可称他为 Linus 或李纳斯，正是这位来自芬兰的天才，在 1991 年 1 月 2 日，攥着在圣诞节和生日得到的钱，偶然地做出了一个重大的财政决定，分期三年买一台价格 3500 美元得相貌平平得计算机，从而 Linux 开始了。

企鹅则是 Linux 的标志，很多人可能不知道 Linus，但是却可能知道这只企鹅，这是一个奇怪的现象，就像很多人知道微软，却不知道比尔盖茨。不管怎么说，是 Linus 塑造了这只企鹅，并让它有一副爽透了的样子，就像刚刚吞下一扎啤酒。除此之外，这只企鹅还要很特别，其他的企鹅都是黑嘴巴黑脚蹼，但它却是黄嘴巴黄脚蹼，这使它看上去好像是鸭子与企鹅的杂交品种，也许它是唐老鸭在南极之旅中与一只当地企鹅一夜倾情的结晶。

其次，在您继续之前，我还想请您问自己一个问题：我在强迫自己学习内核么？我很希望您能回答不是，但希望与现实往往都有段不小的距离，因为很多时候，我都发现身边的人是因为觉得内核很高深而强迫自己喜欢。强迫自己去喜欢一个人是多么痛苦的事情。或许，针对这个问题，最让人愉悦的回答是“说实话，我学习的热情从来都没有低落过。”正如 Linus 在自己的自传《Just for Fun》中希望的那样。

Linux 诞生记

1987 年

MINIX 诞生，而我也已端坐于学堂之中，隐去一身的稚气，能够摇头晃脑的吟诵几句诗赋了。若真是冥冥中自有定数的话，或许这时就暗定了 4 年后 Linux 的诞生。

1991 年

Linus Torvalds，一个芬兰的大学生，对于他不能按照意愿访问大学 UNIX 服务器而感到很愤怒，于是开始为一个以后被称为“Linux”的内核而工作，并于这一年的 10 月 5 日发布了 Linux 0.01。

1992 年

4 月，第一个 Linux 新闻组“comp.os.linux”建立。10 月，第一个可以安装的 Linux 版本 SLS 发布。同年，我拿到了平生的第一个毕业证。

1993 年

8 月，第一本关于 Linux 的著作《Linux Installation and Getting Started Version 1》出版。而

这一年，我最敬佩的语文老师患病离去了，从此，我知道了生活中不仅仅只有欢聚，还有伤别。

1994 年

Linux 1.0 发布，并采用 GPL（GNU General Public License，通用公共许可证）协议。大家要 Linus Torvalds 想一只吉祥物，Linus 突然想到小时候去动物园被一只企鹅追着满地打滚，还被咬了一口！既然想不到其它的吉祥物了，干脆就以这支企鹅来当吉祥物算了！

泰坦尼克的狂潮

1995 年

4 月，召开首届 Linux 博览会，一个以 Linux 为特征的商业展览博览会。几个月后，我迎来了第二个中学阶段。

1996 年

Linux 2.0 发布，它第一个支持了 SMP（对称多处理器）架构。此时 Linux 的全球用户已经达到了 350 万左右。

1997 年

首例 Linux 病毒“Bliss”被发现。电影《泰坦尼克号》所用的 160 台 Alpha 图形工作站中，有 105 台采用了 Linux。

1998 年

1 月，第一份 Linux 新闻周刊出版，同时，Netscape 宣布他们将在自由软件许可协议下发布浏览器的源代码，这为 Linux 和自由软件的发展提供了广阔空间。

2 月，Eric Raymond 和他的朋友门提出了“open source”的概念，申请了该商标特权并且组建了 opensource.org 网站，从而开始推动 Linux 的商业化发展。

4 月，Linux 广泛被美国国家公共新闻广播报道，标志 Linux 在主流、非技术性的媒体界首次出现。

5 月，Google 搜索引擎开始流行，不仅仅是因为它是最好的搜索引擎，而且还因为它是基于 Linux 和具有 Linux 特色的搜索网页。

6 月，“从来没有一个用户向我提起 Linux，Linux 就像众多的免费产品一样，虽然它是很小的，却得到了一群忠诚的拥护者。”比尔盖茨在 6 月 25 日的《PC 周刊》上说。

7 月，KDE 和 GNOME 的桌面之争在其拥护者之间愈演愈烈，Linus 以实际行动表明 KDE 非常好用，在这种情况下，KDE1.0 诞生了。Oracle、Informix、Sybase 都宣布将积极支持 Linux。Linux 开始成为一个家喻户晓的词。

9 月，Dave Whiting 和 Dwight Johnson 创建了 LinuxToday.com，该网站后来被 Internet.com 收购，不过它一直是访问量最高和最容易阅读的 Linux 入门网站。

12 月，一篇来自 IDC 的报导说 Linux 的发行量在 1998 年涨了 200% 以上，它的市场占有率也增加了 150% 以上。Linux 拥有 17% 的市场占有率并且增长率超过了市场上其它任何一个系统。

同年，我迎来了人生中一个非常重要的时刻：我上大学了！

提前发生的革命

1999 年

1 月, “Linux 2.2 已经发布, 我终于可以松口气了” 创造者 Linus Torvalds 说。

3 月, 首届 LinuxWorld 讨论会和博览会在加洲的圣何塞举行, 作为 Linux 第一个大的商业化的贸易展示活动, 它无疑向世界昭示了 Linux 的到来。

8 月, SG 宣布了与 Red Hat 的合作关系, 并且开始大规模的为内核的发展做贡献。Red Hat 进行了首次公开募股, 股价马上涨到了 50 美元, 在那个时候这个价似乎很高。摩托罗拉公司与 Lineo 建立了合作关系, 进入 Linux 领域并提供嵌入式系统产品, 支持和培训服务。Sun 宣布了 Sun 公共源许可 (Sun Community Source License) 下发行 StarOffice 和开发一个网络版本的办公套件。

9 月, Red Hat 的股票达到了 135 美元, 这个价格在那个时候似乎是难以置信的高。

10 月, Sun 宣布它将在 Sun 公共源许可下公布 Solaris 的源代码。

12 月, VA Linux Systems 的首次公开募股价格是 30 美元/股, 这个价格很快涨到了 300 美元, 它在 NASDAQ 历史上创造了最高的首次公开募股价格。

这一年, 网络进入了宿舍, QQ、mud 等也进入了我们的生活。

2000 年

1 月, VA Linux Systems 宣布创建我们非常熟悉的 SourceForge, 到去年底, SourceForge 已经接到了超过 12000 个项目, 拥有 92000 个注册的开发者。

2 月, 最近的 IDC 报告显示 Linux 现在成为“服务器电脑上第二个最受欢迎的操作系统”, 在 1999 年占了 25% 的服务器操作系统销售额, Windows Nt 为 38%, 占第一位, NetWare 为 19%, 排名第三, IDC 以前曾预测过 Linux 将在 2002 或 2003 年到达第 2 位, 这场革命提前发生了。

3 月, 嵌入式 Linux 协会 (Embedded Linux Consortium) 成立。

8 月, HP、Intel、IBM 以及 NEC 宣布开放源代码发展实验室 (OSDL, Open Source Development Lab) 成立。

9 月, Trolltech 发布了 GPL 下的 Qt 库。

11 月, IBM 宣布将在 2001 年投资 10 亿美元在 Linux。首部基于 Linux 的手机 IMT-2000 在韩国发布。

这一年的某一天, 和同学坐在学校四大发明广场上观看同一首歌演出, 困意盎然, 期间那个粗犷的名歌星的一句话却惊醒了我: “希望你们交通大学为中国的交通事业做出更大的贡献”, 大意如此, 我顿时无语, 他的语言竟然和他的外表一样粗犷。

这一年的暑假, 我第一次来到江南, 在西湖断桥对面的饭馆里, 透过落地窗恰恰看到湖里荷花的位置, 要了份西湖醋鱼和一瓶啤酒, 坐到下午四点钟, 然后顺着苏堤白堤静静的走下去, 直到绕湖一周再次回到断桥, 已是晚上八点, 坐在湖边的长凳上, 一夜无语。

和平、爱情和 Linux

2001 年

1 月, 期待已久的 Linux 2.4 发布。

3 月, Linux2.5 内核高级会议在加州圣何塞举行, 它或许是历史上 Linux 内核 hacker 最完整的一次聚会。

4月，IBM 在几个城市鼓吹“和平、爱情和 Linux”（Peace, Love and Linux）时遇到了麻烦。

6月，Sharp 宣布基于 Lineo 嵌入式系统的 Linux PDA 即将上市。

这一年底，找工作的季节，我深刻认识了 IT 泡沫和 9.11，找所谓的好工作无门和出国无门，我无奈选择考研。

2002 年

Linus Torvalds 将 Linux 2.4 交由巴西 18 岁的内核开发人员 Marcelo Tosatti 维护，自己则带领 Linux 2.5 的开发工作。

这一年，我从一个交大到了另一个交大，这个转变似乎很平淡，并不深刻。

Ubuntu 4.10

2003 年

1月，NEC 宣布将在其手机中使用 Linux，代表着 Linux 成功进军手机领域。

6月，IDC 分析师称，2003 年 Linux 服务器在西欧的销售量将达到 18.2 万台，到 2007 年，销售量将增至这个数字的三倍，销售收入将翻一番，达到 19 亿美元。

8月，韩国国家航空公司和 IBM 联合发布声明，表示韩国航空公司将把该公司的核心业务移植到 IBM 的 eServer 服务器当中完成，其中操作系统则采用 Linux。

9月，三星在推出了首款基于 Linux 系统平台的 CDMA 智能手机 SCH-i519。

11月，Linux 2.6 发布，它被认为是第一款真正意义上的企业级内核，这是 Linux 内核从 2001 年以来第一次的大改动。

这一年，我第一次在电视直播里看着自己喜欢的米兰夺得了冠军杯。

2004 年

1月，X.Org 基金会成立。

2月，Linux 标准 2.0 出台，规范了所有能被称为 Linux 操作系统所应该有的特性。

5月，基于 Linux 的路由系统出现。

10月 20 日，Ubuntu 首个版本发布，在五年后的今天 Ubuntu 已经是 Linux 桌面发行版的一个成功典范。

11月，Firefox 1.0 发布，它成为大众关注的焦点，IE 降低了 1 个点的市场份额——像这种事已经多年没有发生过了。Firefox 已经成为了微软 IE 的强有力的对手。

又到了找工作的季节，宣讲会、笔试、面试，我就要离开学校了么？

2005 年

10月，Firefox 的下载量突破了 1 亿大关，这表明，只要产品好，开放源代码软件也能够获得普通用户的青睐。

11月，Sun 开放了除 Java 之外的几乎所有软件，这使得它在一夜间成为了最大的开放源代码软件厂商之一。

12月，Red Hat 公布了第三季度业报，销售收入增长了 43.6%，利润增长了 114%。

这一年夏天，遭遇了到目前为止最为严重的一次失窃，除了 IQ 卡，所有的卡都随着钱夹子消失了，到工行补办牡丹卡时，那慵懒的上海女人说，必须要上海土生土长的本地人来担保，仅仅拥有上海户口的人是不行的。

Richard Stallman 的征婚启事

2006 年

6 月，自由软件之父 Richard Stallman 在自己的网站 <http://www.stallman.org/> 上发布了一则“征婚启事”。

I'm a single atheist white man, 52, reputedly intelligent, with unusual interests in politics, science, music and dance.

I'd like to meet a woman with varied interests, curious about the world, comfortable expressing her likes and dislikes (I hate struggling to guess), delighting in her ability to fascinate a man and in being loved tenderly, who values joy, truth, beauty and justice more than "success"—so we can share bouts of intense, passionately kind awareness of each other, alternating with tolerant warmth while we're absorbed in other aspects of life.

My 22-year-old child, the Free Software Movement, occupies most of my life, leaving no room for more children, but I still have room to love a sweetheart. I spend a lot of my time traveling to give speeches, often to Europe, Asia and Latin America; it would be nice if you were free to travel with me some of the time.

If you are interested, write to rms at stallman dot org and we'll see where it leads.

我，单身，无神论者，白人，52 岁，据说比较聪明，对于政治、科学、音乐和舞蹈有着不同寻常的兴趣。

我想寻找这样一位女士：爱好广泛，对世界充满好奇心，能够清晰表达她的爱憎（我痛恨动脑筋猜测），乐于使男人着迷，渴望被温柔地爱，对于快乐、真理、美和正义的评价高于“成功”。这样的话，我们就能不断对另一方产生热烈而又美好的了解，当我们被生活中其他东西吸引的时候，彼此就能感到宽容的温暖。

我有一个 22 岁的孩子——自由软件运动——他占据了我大部分的生活，没有精力再抚养更多的孩子了，但是我仍然会投入的爱我的爱人。我有大量时间花在巡回演讲上，经常要去欧洲、亚洲和拉丁美洲。如果你有空在某些时间陪我一起旅行，那就最好了。

如果你有兴趣的话，请写信到 rms@stallman.org，让我们看看会有什么结果。

7 月，Ubuntu 被授予 PC World 2006 World Class Award，证明了 Ubuntu 成为 2006 年世界最好的 100 个产品之一。Ubuntu 越来越显示出他的不凡实力，虽说他是免费的，但是后台却是商业公司 canonical，加上太空人老板的聪明才智，逐渐的开始商业合作，比如和 Sun 合作，对有需要的客户提供 Linux 支持服务。

8 月，Linux 业界另外一位狂人，Linuspire 公司总裁 Kevin Carmony 宣布推出免费版本的 Freespire 1.0，该版本中附带有二进制的商业硬件驱动程序，在 Linux 社区中引起轩然大波。27 日，网站 http://linux.inet.hr/poll_filesystem.html 上推出“Your favorite file system?”（你最喜欢的文件系统？）投票活动。

9 月，16 日是“国际软件自由日”（SFD，Software Freedom Day 2006）。

10月，Oracle Unbreakable Linux 发布，Oracle 成为第一个推出自有 Linux 服务的非操作系统软件厂商。17日，FSG（自由标准组，一个非赢利的致力于开发和促进自由开放软件的标准的组织）宣布与 O'Reilly Media 合作，共同为 Linux 应用程序开发人员提供类似 MSDN 的服务，该服务将作为 LSB (Linux Standard Base) Developer Network 的一个组成部分。

11月，微软和 Novell 达成一揽子协议，号称要改善 Linux 和微软操作系统的兼容问题。看着昔日的对手用“+”连起来是否会觉得古怪？



这一年，三次去青岛，回来时遭遇三次严重的飞机晚点，让我疑惑这个世界怎么了？

来自微软的指控

2007 年

1月，虚拟人生游戏（Second Life）客户端开源。两大 Linux 领导社团 OSDL 和 Free Standard Group 宣布合并为新的 Linux Foundation（Linux 基金会），此举将促进社区的资源整合，也使 Linux 在企业市场能够更加高效地参与竞争。

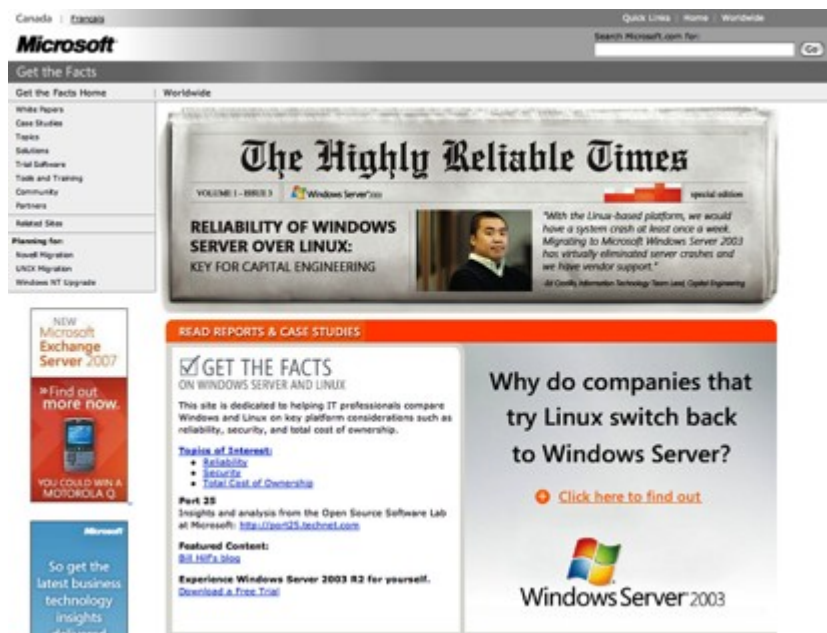
2月，Bill Xu 发起了一个“致招商银行的公开信”的行动，希望用这种方式促使招商银行改变在公众服务中使用专属软件的作法，取消客户端上的 ActiveX 技术，而转用其他公开的、开放的、不限制用户平台的技术。据说，浦发银行的网络银行能很好的支持 Firefox。

3月，Novell 推出模仿苹果的“Mac vs PC”广告，它在广告中插入了第三者：Linux——一位迷人的年轻女子。Novell 用此来宣传预装 Novell Linux 的 PC，一共发布了三个视频，你可以在 www.youtube.com 上看到它们。15日，Novell 公开表示，同意从总费用上说 Linux 比 Windows 要昂贵，这使它在开源社区的名誉进一步恶化。

4月，Dell 推出预装 Ubuntu 操作系统笔记本。

5月，微软声称 Linux 内核侵犯了微软的 42 项专利，而用户界面和其它设计方面也有 65 项侵权，OpenOffice.org 也被指控侵犯 45 项专利，还有 83 项是针对其它免费开源软件。同一个月，微软加拿大网站推出了一个“Get the Facts”（了解真相）页面，赤裸裸地对 Linux 进行了攻击，有趣的是页面上方放置了一张《The Highly Reliable Times》报纸截图，标题模仿《纽约时报》风格。

“报纸”中写道：“我们采用 Linux 平台以后每周至少遭遇一次系统崩溃问题。而迁移到微软 Windows Server 2003 后真正消灭了系统崩溃问题，另外我们还能获得厂商支持。”



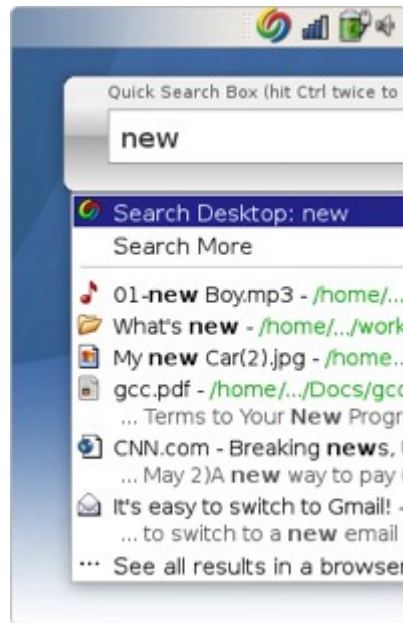
Get the Facts 页面

还是 5 月，Firefox 在 Linux 中显示的表单控件，特别是单选框，比较丑陋的问题被修正。



Firefox 表单控件修正前后比较

6 月，5 日微软和 Linux 发行商 Xandros 宣布，双方达成了一个技术和法律上的合作。Red Hat、Ubuntu 与 Mandriva 拒绝与微软进行专利交易。28 日，Google 桌面搜索 Linux 版正式发布。29 日，第三版 GNU 通用公共许可证 GPLv3 发布。



Google 桌面搜索 Linux 版

7月，Fcitx 小企鹅输入法开源项目终止。做为 Linux 平台上最受欢迎的两大中文输入法之一——Fcitx 小企鹅输入法，在其官方主页上宣布项目终止开发。声明中提到，有“编程高手”质疑其代码风格是项目终止的导火索。

8月，SCO 在控告 Linux 侵犯专利权的官司中败诉，从而申请破产保护。SCO 面市时以 Linux 销售商 Caldera Systems 的面目现身，然后从 Santa Cruz Operation 收购了 Unix 业务，之后重名为 SCO 集团。然后他们放弃了 Linux 业务，并开始起诉 IBM、Novell 及其他公司。他们认为 IBM 破坏了他们签署的 Unix 协议，将 SCO 特有的 Unix 技术在开源的 Linux 社区发布出去。Jim Zemlin 对此评论说：“如果它们把事业基础建立在协助 Linux，而不是攻击 Linux，那么它们大可享受像 RedHat 这些公司一样的成功，而不是沦落到申请破产保护的下场。”

10月，Acacia Research 通过其子公司 IP Innovation 向 RedHat 和 Novell 提出控告，RedHat Linux 操作系统及 Novell 旗下的 SUSE Linux Enterprise Desktop 与 SUSE Linux Enterprise Server 侵犯了他所拥有的专利。随着围绕开源的纠纷不断，2007 年对于律师来说注定是“丰收”的一年。

11月，Google 推出基于 Linux 的开源移动平台 Android。Phoronix 网站发布了 ATI 显卡在 Linux 和 Vista 下的游戏性能对比测试，结果令人鼓舞，在 Linux 下的游戏性能首次超越了 Windows!

这一年，我开始在 blog.csdn.net/fudan_abc 上和两位伙伴写《Linux 那些事儿》。

首款 Android 手机

2008 年

1月，Nokia 宣布收购了著名开源跨平台开发工具 Qt 的开发商 Trolltech。

2月，Google 资助 Linux 版 Photoshop 的研究。

4月，Sun 移除了 Java 最后的限制，将其彻底开源。

7月，腾讯公司在这个月的最后一天发布了 QQ for Linux 1.0 Preview 版，这是第一次官方的版本。

9月，Google 联合 T-Mobile、HTC，正式发布了首款 Android 平台的手机 G1。Google 开源浏览器 Chrome 发布，发布仅仅几个小时，它的总体占有率就达到了 2%。

10月，OpenOffice3.0 发布，这对 Linux 的普及和实用化影响巨大。月底 Ubuntu 8.10 发布，Fedora 10 发布。

12月，各类发行版的 Linux 操作系统占据了大约三成的上网本市场份额。

对于我来说，这一年的基调是出差，大半年之后回到上海，很多地方很多事情都变得陌生起来，才发觉忘却其实也是一件很容易的事情。

Linux 信用卡

2009 年

1月，Linux 兼容内核正式使用项目 Unix 名称 Longene，中文别名“龙井”。兼容内核是一个自由、开源的操作系统项目，目的是要把 Linux 的内核扩充成一个既支持 Linux 应用、也支持 Windows 应用，既支持 Linux 设备驱动、也支持 Windows 设备驱动的兼容内核，使用户可以直接在 Linux 操作系统上高效运行 Windows 应用。

2月，微软起诉 GPS 设备制造商 Tomtom 侵犯其 8 项专利权，Tomtom 的 GPS 设备采用的是 Linux 系统，尽管微软声称 Linux 侵犯其专利期已有多年，但该案被视为微软状告 Linux 侵权的第一案。

3月，Adobe Reader 9.1 for Linux 发布。UltraEdit 正被移植到 Linux，名为 UEX，意即 UltraEdit for Linux。

4月，IDC 最新发表的题为《Linux 在新经济中的机会》的报告称，用户 2009 年的 Linux 开支预计将比 2008 年增长 21%，超过整个软件市场的增长速度。整个软件市场 2009 年的增长率是 2%。

5月，Nokia 宣布开放 Qt 源代码仓库，以便让社区的开发者能够进一步参与 Qt 的开发。

6月，法国的 ENAC 开发组为 Linux 内核开发了类似 iPhone 的 Multi-touch（多点触摸）技术支持。

7月，Linux 基金会与 CardPartner 和 UMBrella 银行共同推出了 Visa 白金信用卡，正面印有 Linux 吉祥物。每办理一张这样的信用卡就可以为 Linux 基金会带来 50 美元的赞助，使用该卡每消费一次 Linux 基金会就能从中获得 1% 的金额。



Linux 信用卡

8月，微软在提交给美国证券交易委员会的年度文件中，将Unbutu列入竞争对手。

9月，目前为止最新的内核版本 2.6.31 发布，Linux 成为首个正式支持 USB3.0 的操作系统。

10月，Ubuntu 9.10 发布。

11月，Vim的作者Bram Moolenaar推出了新的编程语言Zimbu，一种不拐弯抹角直截了当的实验性编程语言。Moolenaar表示Zimbu集现有语言的优点于一身，同时避开它们的不足。Zimbu代码清晰易读，使用范围广泛——既能写OS kernel，又能写脚本，还能写大的GUI程序，可以编译和运行在几乎所有系统上。

关于本书：

本文节选自《Linux内核修炼之道》。本书作者任桥伟，CU论坛网名ilthv.cn，博客地址是http://blog.csdn.net/fudan_abc/，另外他还是《Linux那些事儿》系列书籍作者之一。欢迎大家就相关技术问题和取得联系。

本书将Linux内核的学习分为四个层次：全面了解，掌握基本功；兴趣导向，选择重点深度钻研；融入社区，参与开发做贡献；坚持，坚持，再坚持。总结起来，就是“全面了解抓基本，兴趣导向深钻研；融入社区做贡献，坚持坚持再坚持。”（如果您是一个修真小说爱好者，尽可以将其与炼气、筑基、结丹和元婴等层次相对应。）

第一层次修炼的内容包括了前三章，目的是希望您能够对Linux以及内核有个全面的认识 and 了解，掌握分析Linux内核源代码的分析方法。

第1章主要介绍了Linux的18年成长史，或许您会乐意陪我一起缅怀下这过去的十八年。

第2章介绍内核的配置和编译过程，和任何大型软件源码的学习一样，学会编译和配置是第一步。

第3章介绍学习内核需要的基础，内核的体系结构、目录结构、代码特点，浏览内核代码的工具，最后，突出强调了内核源码分析过程中极为重要的两个角色——Kconfig和Makefile，并以USB子系统为例，演示了如何利用这两个角色进行代码分析。

第4章讨论系统的初始化，万事开头难，系统的初始化是一个很复杂的过程，不过对于内核源码的学习来说，以这个部分开始应该是个不错的选择。特别是子系统初始化的讨论，应该是您选择任何内核子系统开始分析时都需要了解的内容。

第 5 章讨论系统调用，它是应用程序和内核间的桥梁，学习并理解它是我们走向内核的一个很好的过渡。

第 6 章讨论内核的中断处理机制，包括几乎任何一本内核书籍都没有涉及的通用 IRQ 层。

第 7 章讨论进程的内存抽象，以及进程如何被创建和销毁。

第 8 章讨论进程的调度，重点讨论了在内核历史上具有重要地位的 O(1)调度器和最新的 CFS 调度器。

第 9 章讨论内存管理，内存就是进程的家，这里讨论内核如何为每个进程都分配一个家，并尽可能的去做到“居者有其屋”，以及保证每个家的安全。

第 10 章讨论文件系统，主要是虚拟文件系统（VFS），它通过在各种具体的文件系统之上建立一个抽象层，屏蔽了不同文件系统间的差异。

第 11 章讨论设备驱动，对于驱动开发来说，设备模型的理解是根本，spec、datasheet 与内核源代码的利用是关键。

第 12 章讨论参与内核开发需要了解的一些基础信息。

第 13 章讨论了内核的调试技术，与第 12 章一样，您可以仅仅将这些内容看成内核修炼中的一些 tips。

革命尚未成功，我等仍需努力。——与君共勉之。

Linux 上防火墙 tcp_wrappers 详解

ChinaUnix 网友: Sandy

今天我来为大家讲一下 linux 上一种特殊的防火墙 TCP_wrappers 防火墙, TCP 封包会先经过所谓的 IP 过滤机制(IP Filtering), 这是 Linux 提供的第一层保护, 他可以将你不想要的来源 IP (经由 TCP 封包的 Head 资料)先当掉再说! 如果可以通过的话, 在就是要通过 TCP_wrappers 过滤.如果上面两个都通过了,再就根据每个服务访问控制的设定决定客户机能得到不同的权限和信息。

TCP_wrappers 防火墙主要涉及到两个文件/etc/hosts.allow 和/etc/hosts.deny,这两个文件被整合到 xinetd 中去了.我们需要安装 tcp_wrappers 这个套件,因为这两个文件本身就是 tcp_wrappers 的设定文件.也就构成了一个基础的防火墙。

tcp_wrappers 设定 tcp 封装的包是否可以进入/etc/hosts.allow 和/etc/hosts.deny 中,如果一个服务是受到 xinetd 或 TCP_Wrappers 的控制时, 那么该服务就会受限于 hosts.allow 与 hosts.deny 的管理了! 你的首先判断一下某个服务是否可以使用 tcp_wrappers 防火墙.vsftpd . telnet.sendmail、sshd、tcpd、xinetd、gdm、portmap 都可以使用。很多服务在/etc/xined.d/目录中,可以对服务进行规则的设置...查看一个服务是否可以用 tcp_wrapeer 控制.用此命令#ldd `which servername`.返回信息中有很多 lib 开头的链接文件..那些是库文件,说明此服务受 TCP_Wrapper 规则控制...

先看一下#cat /etc/xinetd.conf | less,如果 enable 和 disable 都为 yes,那就以拒绝优先..

only_from #定义只有哪些主机可以访问该服务.

no_access #定义哪些主机不能访问该服务.

当 only_from 和 no_access 定义的主机由包含的关系时,以小范围的为主,比方说,我在 only_from 中

定义运行 1.1.1.0 网段主机可以访问,在 no_access 中定义 1.1.0.0 主机不能访问.这时而不是拒绝优先了,切忌.而是以 only_from 为准...

cps #定义限制连接进来的主机数目.

per_source #定义一台主机最大连接数目,通常和 cps 混用...

instance #定义最大连接数目,这个数目是指不同的主机连接进来的数目,而不是同一台主机连接进来的数目..

bind #定义指定监听的 IP 地址(后只跟 IP 地址,而不是网卡的名字.)

baner #定义欢迎信息,后面指定一个文件路径.此文件内容可自己定义..

socket_type = stream #定义使用 tcp 协议..

single_threaded #定义单线程

multi_threaded #定义多线程

我们以 ftp 服务和 telnet 为例,有一点得说明,机器名不只是/etc/sysconfig/network 中定义,还有/etc/hosts,这两个文件中的机器名必须一致,否则即使你开启了 telnet,别人也无法 telnet 进去...当时我就犯了这个错误.让我郁闷好长时间,终于发现问题所在..我服务器 IP:192.168.0.195 先安装一个 xinetd 包,然后安装 ftp 包和 telnet 包.接下来我们开始看/etc/hosts.allow 文件..

先挂载光盘(过程略)然后安装相应的包...

```
[root@station195 Server]# rpm -ivh xinetd-2.3.14-10.el5.i386.rpm
```

```
[root@station195 Server]# rpm -ivh vsftpd-2.0.5-16.el5.i386.rpm
```

```
[root@station195 Server]# rpm -ivh telnet-0.17-39.el5.i386.rpm
```

```
[root@station195 Server]# rpm -ivh telnet-server-0.17-39.el5.i386.rpm
```

编辑/etc/xinetd.d/telnet 将 disable = yes 改为 disable = no(启动 telnet,默认不会开启.)

```
[root@station195 Server]# service xinetd restart
```

```
[root@station195 Server]# chkconfig xinetd on
```

编辑 hosts.allow 文件,写入一行 vsftpd:192.168.0.0/255.255.255.0 EXCEPT 192.168.0.192

再编辑/etc/hosts.deny 文件,加入一行 vsftpd:ALL,然后执行此命令

```
[root@station195 ~]# chkconfig --level 35 vsftpd on
```

```
[root@station195 ~]# service vsftpd restart
```

当我们用 192.168.0.192 机器来访问服务器时发现访问不了.原因在于当客户端进行 TCP 请求时服务器首先读取 hosts.allow 文件.匹配 192.168.0.0 网段内所有主机除了 192.168.0.192 外...所以里面没有匹配到 192.168.0.192, 然后会读取 hosts.deny 文件 vsftpd:ALL 这个 ALL 中包含有客户端的 192.168.0.192 然后就被拒绝了.当换个 IP(必须是 192.168.0.0 网段内的主机)再次访问服务器时就可以登录进去了...

我们还可以让别人登录服务器时的消息发给 服务器,这就用到 spawn 命令....编辑 hosts.allow 文件....写入一句 in.telnetd:ALL:spawn /bin/echo `date` %c %d | /bin/mail -s "somebody access our ftp." root

意思是说当有主机访问服务器时会向管理员发封邮件内容就是 somebody access our ftp ...

%c 提取客户机的信息,%d 是守护进程的名字...当我们用客户机来 telnet 服务器后,服务器自动向自己发个邮件.我们可以在服务器上使用 mail 命令查看内容...

From root@station195.example.com Thu Feb 25 14:35:47 2010

Date: Thu, 25 Feb 2010 14:35:47 +0800

From: root <root@station195.example.com>

To: root@station192.example.com

Subject: somebody access our telnet.

Thu Feb 25 14:35:47 CST 2010 192.168.0.192 in.telnetd

我们还可以给对方造成一种假象,提示对方输入用户名和密码,输入都正确,但是进不去...这个会让人很郁闷是不是很好玩啊....下面就要用到 twist 命令了..我教你怎么玩...so easy..就那 ftp 为列...

编辑 hosts.allow 文件...写入一句

```
vsftpd:ALL:twist /bin/echo "welcome to server."
```

当在次访问 FTP 时就会提示输入用户名和密码,即使输正确了.但是你进不去的...

还可以返回给客户一句话,比方说在 hosts.allow 中写入

```
vsftpd:ALL:twist /bin/echo `date` "connection refused by %s."
```

再次访问 ftp 时,直接会退出来,返回的信息...

```
C:\>ftp 192.168.0.195
```

```
Connected to 192.168.0.195.
```

```
Thu Feb 25 15:15:41 CST 2010 connection refused by vsftpd@192.168.0.195.
```

```
Connection closed by remote host.
```

也可以将 spawn 和 twist 命令一起用...还在 hosts.allow 中插入一句话

```
vsftpd:ALL:spawn /bin/echo `date` %c to %s denied. >>/var/log/tcpwrapper.log:twist /bin/echo "attempt log to %s failed."
```

当登录服务器时会将返回想信息定向到 tcpwrapper.log 中..

```
Thu Feb 25 15:32:44 CST 2010 192.168.0.200 to vsftpd@192.168.0.195 denied.
```

```
C:\>ftp 192.168.0.195
```

```
Connected to 192.168.0.195.
```

```
attempt log to vsftpd@192.168.0.195 failed.
```

```
Connection closed by remote host.
```

我们还可以定义欢迎信息...banners,这儿就不详细讲述了很简单.这儿我讲 hosts.allow 文件最多.因为很多规则我们可以在这一个文件中 定义.系统会先读取/etc/hosts.allow 文件内容,当匹配不到规则的时候就会读取 hosts.deny 文件的内容了.建议初学者写规则时分在 两个文件中写,熟练后再写在一个文件里...TCP_wrapper 很重要,会经常用到那两个配置文件做访问控制用,喜欢 linux 的朋友要学好了。

PHP 应用提高篇

摘要: 本文档的阅读对象是有一定经验的 PHP 程序员,文档中将介绍在 PHP 开发中的一些开发技巧,希望能对读者有一定的启发。

1. 提高 PHP 的运行效率

PHP 的优点之一是速度很快，对于一般的网站应用，可以说是已经足够了。不过如果站点的访问量很高、带宽窄或者其它的因素令服务器产生性能瓶颈的时候，你可能得想想其它的办法来进一步提高 PHP 的速度了。

1.1. 代码优化

- 1、用 `i+=1` 代替 `i=i+1`。符合 C/C++ 的习惯，效率还高。
- 2、尽可能的使用 PHP 内部函数。自己编写函数之前要详细查阅手册，看有没有相关的函数，否则费力不讨好。
- 3、能使用单引号字符串尽量使用单引号字符串。单引号字符串的效率要高于双引号字符串。
- 4、用 `foreach` 代替 `while` 遍历数组。遍历数组时 `foreach` 的效率明显高于 `while` 循环，而且不需要调用 `reset` 函数。两种遍历方法如下：

程序 1：

```
reset ( $ arr);  
while (list( $ key, $ value) = each ( $ arr)) {  
    echo "Key: $ key; Value: $ value  
    n";  
}
```

程序 2：

```
foreach ( $ arr as $ key => $ value) {  
    echo "Key: $ key; Value: $ value  
    n";  
}
```

1.2. 压缩页面

HTTP1.1 协议支持页面压缩传送，也就是说服务器把一个页面压缩传送到客户端，然后在客户端将页面解压缩再显示给客户。在服务器端有两种传输方式，一种是页面事先已经压缩好了，传送时只要将压缩页面传送到客户端就行，这种适用于静态网页多的情况，但是对于大多数站点，动态页面比较多，这种方法不太适合，因为很多传到客户端的页面其实是没有的，是服务器接到客户端用户请求动态产生的，所以就要求每生成一个动态页面都要在传到客户端以前先打包压缩。从 PHP 的 4.0.4 版以后，可以在 `php.ini` 文件中增加一行配置 “`output_handler = ob_gzhandler`”，这样每个动态生成的页面在传送到客户端之前都会进行压缩，但是根据 PHP 官方站点的说明，这个参数不能与 “`zlib.output_compression = on`” 参数同时使用，因为容易引起 PHP 工作不正常，另外它只能压缩 PHP 程序的动态生成的页面，对于大量的静态页面尤其是图像文件就不行了。但是 `mod_gzip` 模块为 Apache 提供了将静态页面在传给客户端以前先压缩的功能，它的压缩比最大能到 10，一般情况下可以到 3，也就是说网站的传输速率一下提高了三倍多。要使用 `mod_gzip` 还要对 Apache 进行相应的配置，需要在 `httpd.conf` 文件中加入一些参数：

```
mod_gzip_on Yes (模块是否生效)  
mod_gzip_minimum_file_size 1002 (最小压缩文件大小)  
mod_gzip_maximum_file_size 0 (最大压缩文件大小, 0 表示没有限制)  
mod_gzip_maximum_inmem_size 60000 (最大可占用内存)  
mod_gzip_item_include file "..gif102SINA>DOUBLE_QUOTATION (以 gif 结尾
```


的文件要压缩传送)

```
mod_gzip_item_include file ".txt"102SINA>DOUBLE_QUOTATION
mod_gzip_item_include file ".html"102SINA>DOUBLE_QUOTATION
mod_gzip_item_exclude file ".css"102SINA>DOUBLE_QUOTATION
```

1.3. 文件缓存

这种方法通常是针对 PHP、PERL 等 CGI 程序而言的，因为这些程序有一个共同的特征就是接到用户的请求后不是马上将结果返回给用户，而是经过解释器解释执行后将执行结果返回给客户，这期间通常都要涉及到数据库的访问。这样就会出现一个问题，当两个用户访问同一个页面时，系统将分别对两个请求进行操作，但事实上这两个操作可能是一模一样的，这样无形当中增加了系统的负担。所以通常的解决办法是在系统内存中开辟出一段空间，当用户第一次访问页面后将执行结果存放在该内存中，当有用户再一次访问该页面时，系统就将页面直接从内存中调出而不需要重新解释执行，这段内存空间就叫缓存。目前流行的缓存管理程序是 Zend Technologies 公司的 Zend Cache。

2. 执行系统外部命令

PHP 作为一种服务器端的脚本语言，象编写简单，或者是复杂的动态网页这样的任务，它完全能够胜任。但事情不总是如此，有时为了实现某个功能，必须借助于操作系统的外部程序（或者称之为命令），这样可以做到事半功倍。

在 PHP 中调用外部命令，可以用如下三种方法来实现：

2.1. 用 PHP 提供的专门函数

PHP 提供共了 3 个专门的执行外部命令的函数：system()，exec()，passthru()。

system()

原型：string system (string command [, int return_var])

system() 函数很其它语言中的差不多，它执行给定的命令，输出和返回结果。第二个参数是可选的，用来得到命令执行后的状态码。

例子：

```
system("/usr/local/bin/webalizer/webalizer");
```

exec()

原型：string exec (string command [, string array [, int return_var]])

exec() 函数与 system() 类似，也执行给定的命令，但不输出结果，而是返回结果的最后一行。虽然它只返回命令结果的最后一行，但用第二个参数 array 可以得到完整的结果，方法是把结果逐行追加到 array 的结尾处。所以如果 array 不是空的，在调用之前最好用 unset() 最它清掉。只有指定了第二个参数时，才可以用第三个参数，用来取得命令执行的状态码。

例子：

```
exec("/bin/ls -l");
exec("/bin/ls -l", $res);
exec("/bin/ls -l", $res, $rc);
```

passthru()

原型：void passthru (string command [, int return_var])

passthru()只调用命令，不返回任何结果，但把命令的运行结果原样地直接输出到标准输出设备上。所以 passthru()函数经常用来调用象 pbmplus (Unix 下的一个处理图片的工具，输出二进制的原始图片的流) 这样的程序。同样它也可以得到命令执行的状态码。

例子：

```
header("Content-type: image/gif");
passthru("./ppmtogif hunte.ppm");
```

2.2. 用 popen()函数打开进程

上面的方法只能简单地执行命令，却不能与命令交互。但有些时候必须向命令输入一些东西，如在增加 Linux 的系统用户时，要调用 su 来把当前用户换到 root 才行，而 su 命令必须要在命令行上输入 root 的密码。这种情况下，用上面提到的方法显然是不行的。

popen()函数打开一个进程管道来执行给定的命令，返回一个文件句柄。既然返回的是一个文件句柄，那么就可以对它读和写了。在 PHP3 中，对这种句柄只能做单一的操作模式，要么写，要么读；从 PHP4 开始，可以同时读和写了。除非这个句柄是以一种模式（读或写）打开的，否则必须调用 pclose()函数来关闭它。

例子 1

```
$fp=popen("/bin/ls -l", "r");
```

例子 2

```
/* PHP 中如何增加一个系统用户
下面是一段例程，增加一个名字为 james 的用户，
root 密码是 verygood。仅供参考
*/
$sucommand = "su --login root --command";
$useradd = "useradd ";
$rootpasswd = "verygood";
$user = "james";
$user_add = sprintf("%s %s %s", $sucommand, $useradd, $user);
$fp = @popen($user_add, "w");
@fputs($fp, $rootpasswd);
@pclose($fp);
```

3. 养成良好的程序风格

在很多情况下 PHP 最可贵的特点也可能是它最薄弱的环节就是它的语法松散性。PHP 能够如此广泛的被使用，因为它使得许多没有经验的 Web 开发者能够制作出强大的应用程序，而用不着过多的考虑规划、连贯性和文档。不幸的是，正是以上的特点，很多的 PHP 源代码很臃肿，很难阅读甚至无法维护。决定代码可维护性的重要因素是代码的格式和注释。一个工程的所有代码应该以贯穿始终的形式组织。以下介绍的是如何在 PHP 程序中养成良好的程序代码风格。

3.1. 缩进

开发者所有的代码应该完全按照缩进的方式书写。这是提高代码可读性最基本的措施。即使你没有注释你的代码，缩进对于让其他人读懂你的代码也是非常大的帮助。

3.2. 增加注释

在编程时添加注释是一个好的习惯。PHP 允许在页面代码中添加注释，具体的注释方法与 C 语言注释语法相同，在脚本中可以加注释。可以用 `"/**" 和 "/**" 来注释一段话。可用双斜杠 “//” 作为注释符。`

3.3. 控制结构

这个很大程度上取决于个人口味。我仍然可以看到很多的控制结构代码不带分支语句造成可读性非常差，如???你使用 IF 语句时不带分支，不但可读性变差，当其他人修改你的程序时，还会造成很多的 bug。请看下例：

不好的例子：

```
if ($a == 1) echo 'A was equal to 1';
```

这是非常难以辨认的。它能够正常工作，但是除了你之外，别人根本不会赞赏这句代码。有改进的例子：

```
if ($a == 1)
echo 'A was equal to 1';
```

现在至少这句代码可以读懂了，但是仍然没有很好的可维护性。如果当 `$a==1` 的时候我希望一个附加的事件发生，或者需要添加分支呢？如果后来的程序员忘记了添加大括弧或者 `else` 关键字，那么程序中将出现 bug。

完美的例子

```
if (($a == 1) && ($b==2)) {
    echo 'A was equal to 1'; //很容易的可以添加其他代码
} elseif (($a == 1) && ($b==3)) { //其他操作
}
```

请注意在 `if` 和 `elseif` 后面的空格，这会将本语句和函数调用区分开来，此外，虽然在 `elseif` 的执行程序段中没有语句，只有注释，表面上显得多余可是却给以后维护程序的程序员给予了非常方便的提示，并且非常利于添加功能。

3.4. 利用 Include 来实现功能模块化

可以将常用的功能函数存于一个 PHP 文件，在其它 PHP 页中要用其中的功能时，将包含函数的 PHP 文件包含在调用函数的 PHP 文件中。这时可使用 `Include` 函数。具体语法为：

```
include($FileName);
```

使用时应该注意：

- 1、应避免自包含，即 `File1` 包含 `File1`；在多个文件中有包含语句时，应避免间接自包含即循环包含，如 `File1` 包含 `File2`，`File2` 包含 `File3`，`File3` 包含 `File1`。
- 2、被包含的脚本语言的类型必须是 PHP 语言类型或脚本语句段。

利用 U 盘安装 Linux 操作系统

ChinaUnix 网友: changzi100

前一刻盘太多，手头没有 DVD 拉，今天又想装一下系统，于是想起以前用过的 grub4dos 来。

基本思想是把 grub4dos 装到 U 盘里，同时将 iso 文件（3G 多）也弄到 U 盘中，这样直接从 U 盘启动安装系统。

说起来是很简单，但是真到实际操作的时候很多问题。下面我来总结一下会遇到的问题和解决方法。

对于装系统来说不同的人有不同的方法，但是最简单的就是用系统盘，放到光驱中直接启动后安装，这个方法想必大家也都知道。这个方法也是我最常用的。

其它的方法从硬盘安装或是从 U 盘安装也好，基本都是利用 grub4dos 引导 Linux 的启动文件 initrd.img 和 vmlinuz。

那好，我就来说说我是如何利用 U 盘启动安装的吧。

1、要让 U 盘可以启动系统。

有两个操作，一个是设置 bios 让主板从 USB HDD 或是 USB ZIP 启动，另一个是将 U 盘制作成启动 U 盘。

关于 HDD 和 ZIP 等见下面说明（来自互联网）：

HDD，HardDiskDrive 的缩写，即硬盘驱动器的英文名。最基本的电脑存储器，我们电脑中常说的 C 盘、D 盘之类磁盘都属于硬盘驱动器，不过一般我们简称为硬盘。现代硬盘容量一般为 40G、80G、120G、160G 等等，目前最高的桌面级硬盘为 500G。硬盘按体积大小可分为 3.5 寸、2.5 寸、1.8 寸等，按接口可分为 PATA、SATA、SCSI 等，PATA、SATA 一般为桌面级应用，容量大，价格相对较低，适合家用；而 SCSI 一般为服务器、工作站等高端应用，容量相对较小，价格较贵，单性能较好，稳定性也比较高。

ZIP 是指一种高储存密度的磁盘驱动器与磁盘，一片 ZIP 磁盘的容量约 100MB，大小则和传统 3.5"软盘片差不多。另有 LS-120 规格的 120MB 高容量磁盘，而且其磁盘驱动器可读取传统 3.5"软盘片，但是其存取速度稍逊于 ZIP。

FDD 接口：这是内置软驱接口，是传统的软驱接口，直接与电脑主板上的软驱接口相连，价格低廉。软盘驱动器就是我们平常所说的软驱，英文名称叫做“Floppy Disk”，它是读取 3.5 英寸或 5.25 英寸软盘的设备。现今最常用的是 3.25 英寸的软驱，可以读写 1.44MB 的 3.5 英寸软盘，5.25 英寸的软盘已经淘汰，很少会见到。

1. ZIP 模式是指把 U 盘模拟成 ZIP 驱动器模式，启动后 U 盘的盘符大多是 A:

2. HDD 模式是指把 U 盘模拟成硬盘模式；特别注意：如果选择了 HDD 模式，那么这个启动 U 盘启动后的盘符是 C:，在对启动分区进行操作时就容易产生很多问题，比如：装系统时安装程序会把启动文件写到 U 盘而不是你硬盘的启动分区！导致系统安装失败。所以请尽量先选择 ZIP 模式。

3. FDD 模式是指把 U 盘模拟成软驱模式，启动后 U 盘的盘符是 A:，这个模式的 U 盘在

一些支持 USB-FDD 启动的机器上启动时会找不到 U 盘，所以请酌情使用。

完成上面的两个操作就可以制作好从可以从 usb 启动的 U 盘拉，第一个操作不说拉（从 865 的主板开始应该都可以从 usb 启动），第二个操作一盘是通过软件来实现，目前应用的软件有 USB00T 1.7，HPUSBFW2.20.sfx.exe 等，这两个软件不支持大的 U 盘，如超过 2G 的 U 盘，而我手头的 U 盘是 8G 的，用 USB00T 制作完后只有 2G，其余的空间使用不了，HPUSBFW 提示不支持大 U 盘，如果把 iso 文件放到硬盘中（不放到 U 盘中）的话，可以使用 USB00T 这个工具，安装系统选择 iso 文件的时候可以指定文件在哪个目录（硬盘）中，我不使用这个方法，既然 U 盘足够大，我想把 iso 文件放在 U 盘中，这么做有什么好处呢？对我来说的好处是可以给多个新机器装系统，其中的硬盘也都是未格式化地，也就不存在可以把 iso 文件放在硬盘中的可能拉。



图 1 USB00T 1.7 启动界面



图 2 USB00T 制作启动 U 盘时选择的模式

在这里说一下啊，如果 U 盘大于 2G，作成启动盘后只剩下 2G 大小了，装完系统想把 U 盘格式化成本来的大小时，选择“用 0 重置参数”即可。

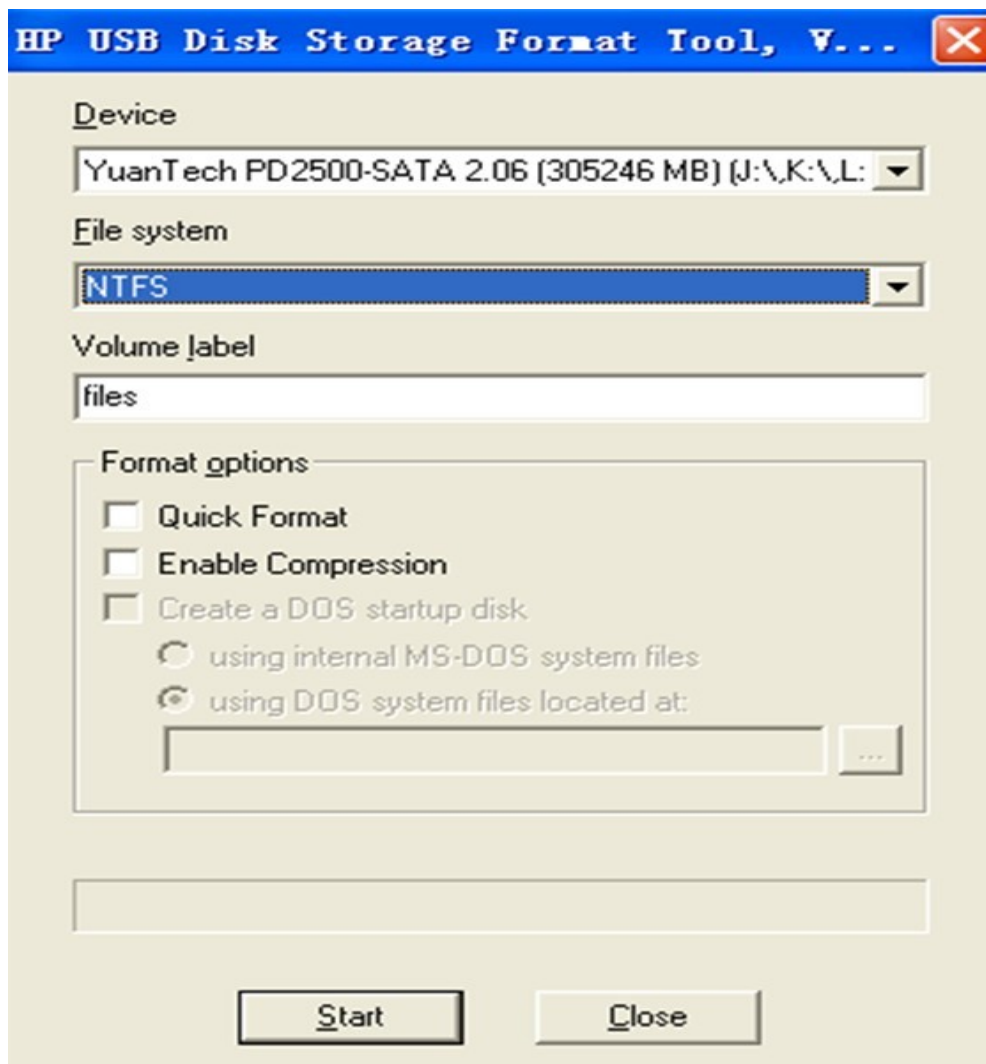


图 3 HPUSBFW2.20 软件启动界面

上面的两个工具不适合我，顺便说一下，使用上面两个工作作启动盘时要手动向 U 盘中复制一些 DOS 文件，或者 HP 的软件可以指定 DOS 的系统文件，此时一定要有 DOS 文件才能制作启动 U 盘，我在网上找了一下，基本上全是 DOS 的镜像，这样的话下载后要用 WINIMAGE 这个软件把镜像中的 DOS 系统文件提取出来。

DiskGenius.exe 这个软件可以实现我的需求，试了一下，还是很好用地。

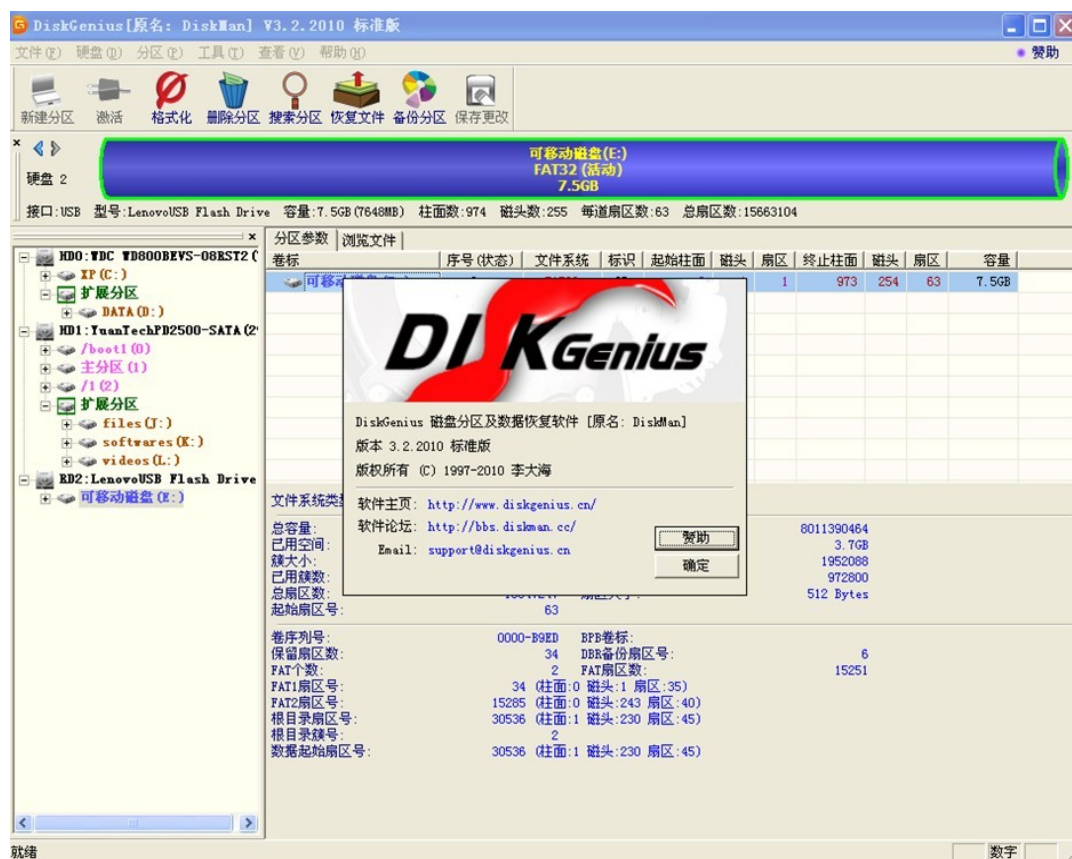


图 4 DiskGenius 软件界面

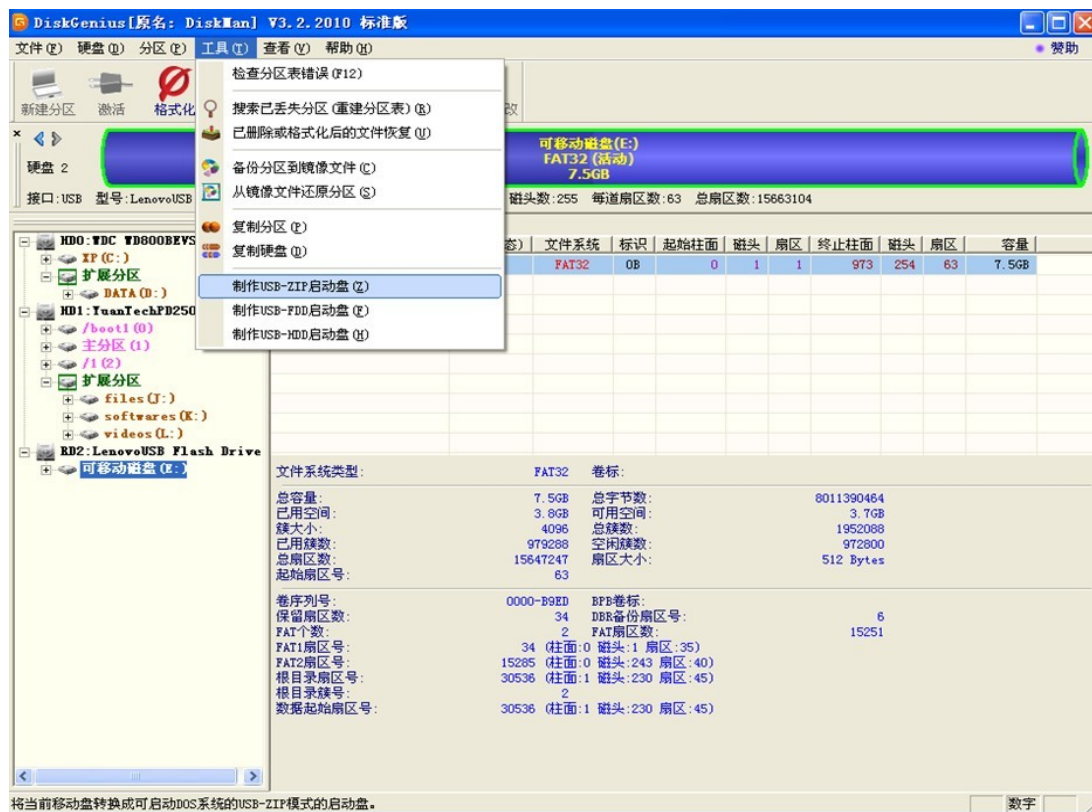


图 5 制作 USB HDD 启动盘操作

用这个软件将 U 盘制作成 USB HDD 启动盘，此时 U 盘可以启动到 DOS 中，在完成操作 1 后可以把它插入到电脑中看看是否启动到了 DOS 中，正常启动后会在 DOS 下进入 DiskGenius，按 ESC 键退出即可，会出现 DOS 提示符 C:\>

这样第 1 步的两个操作已经完成。完成到这步其实可以向 U 盘里装些小工具方便平时维护系统，比如 ghost 之类的，当然，这不是我今天主要的目的。

2、把 grub4dos 装到 U 盘中

这一步需要 grub4dos 这个软件，将其解压缩，找到两个文件：grub.exe、grldr、menu.list，将这两个文件复制到 U 盘的根目录中，此时根目录中已经有很多文件了，其中包括 DOS 系统文件和 DiskGenius 的一些文件，我的目的是装系统，所以 DiskGenius 文件并不需要，但我发现这个软件真的很好噢。

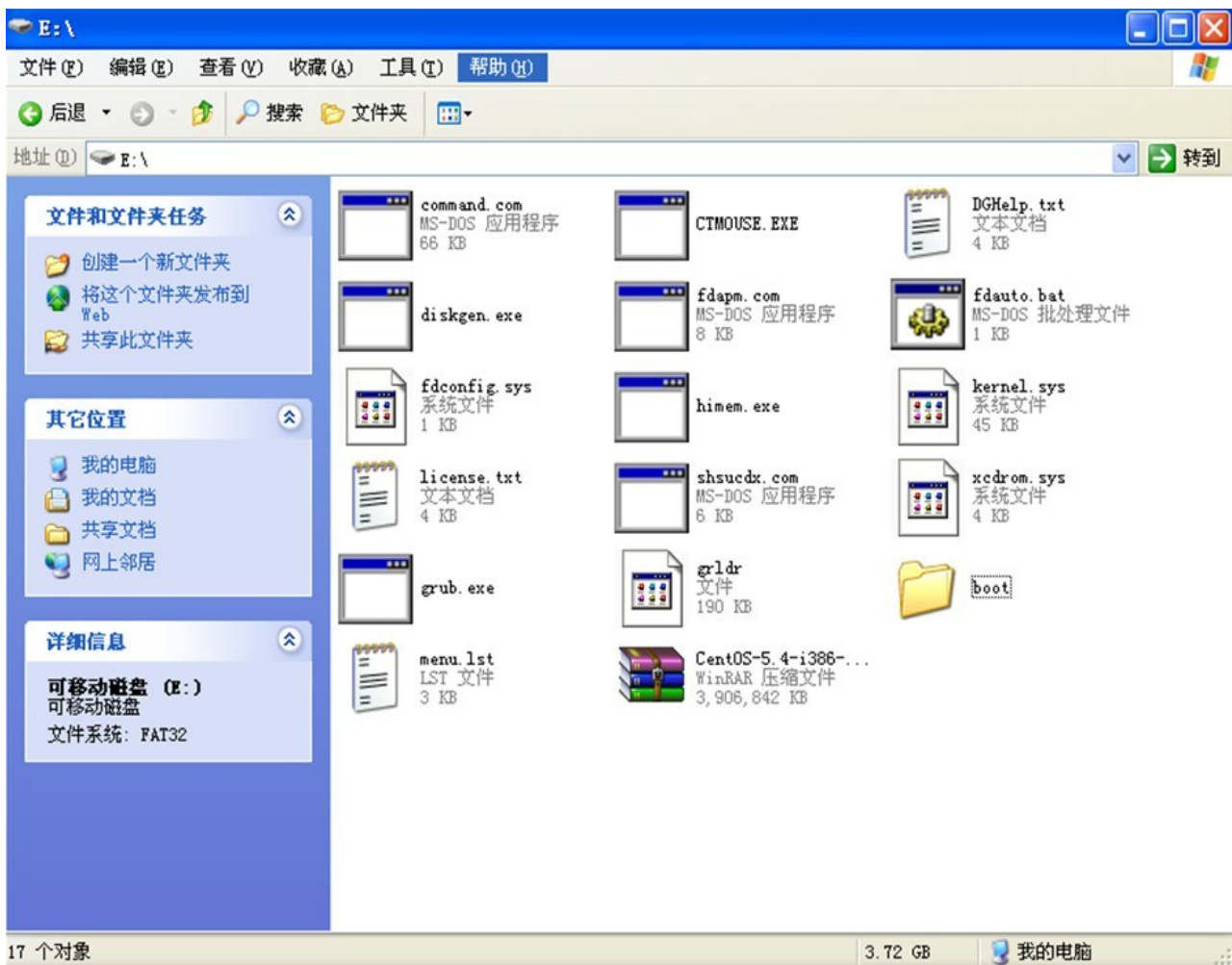


图 6 U 盘中的文件

3、Linux 启动文件

从 iso 中将 vmlinuz、initrd.gz（文件位置 iso/image/boot.iso/isolinux）这两个文件解压缩出来，放在 U 盘中的 boot 文件夹下。

修改 menu.list 文件，在最后面加上下面几行，这几行代表什么意思应该不用多说拉，使用过 Grub 的童鞋都有经验。


```
title Linux Install Setup
```

```
root (hd0,0)
```

```
kernel /boot/vmlinuz root=/dev/hda1
```

```
initrd /boot/initrd.img
```

4、启动电脑

接下来的事儿就是启动电脑，正常的情况是走到步骤 1 的结尾，就是有 DOS 提示符的那里（如下所示），此时输入命令 grub 即可。

```
C:\>grub
```

启动后是 GRUB4DOS 的启动界面，最后一行就有我们修改过的标题 Linux Install Setup。选择这个标题开始 booting。

接下来就是一系列的文字界面对话框，选择语言、键盘类型。

之后出来的对话框是比较有用地，Installation Method，也就是安装方法，有以下几种：

Local CDROM

Hard drive

NFS image

FTP

HTTP

这里我用的是 Hard drive，之前已经将安装的 iso 文件复制到 U 盘中。

接下来是选择 iso 在哪个盘里，根据不同电脑的情况这里就不一样拉，自己看着办喽。

例如：

```
/dev/sda1
```

```
/dev/sda2
```

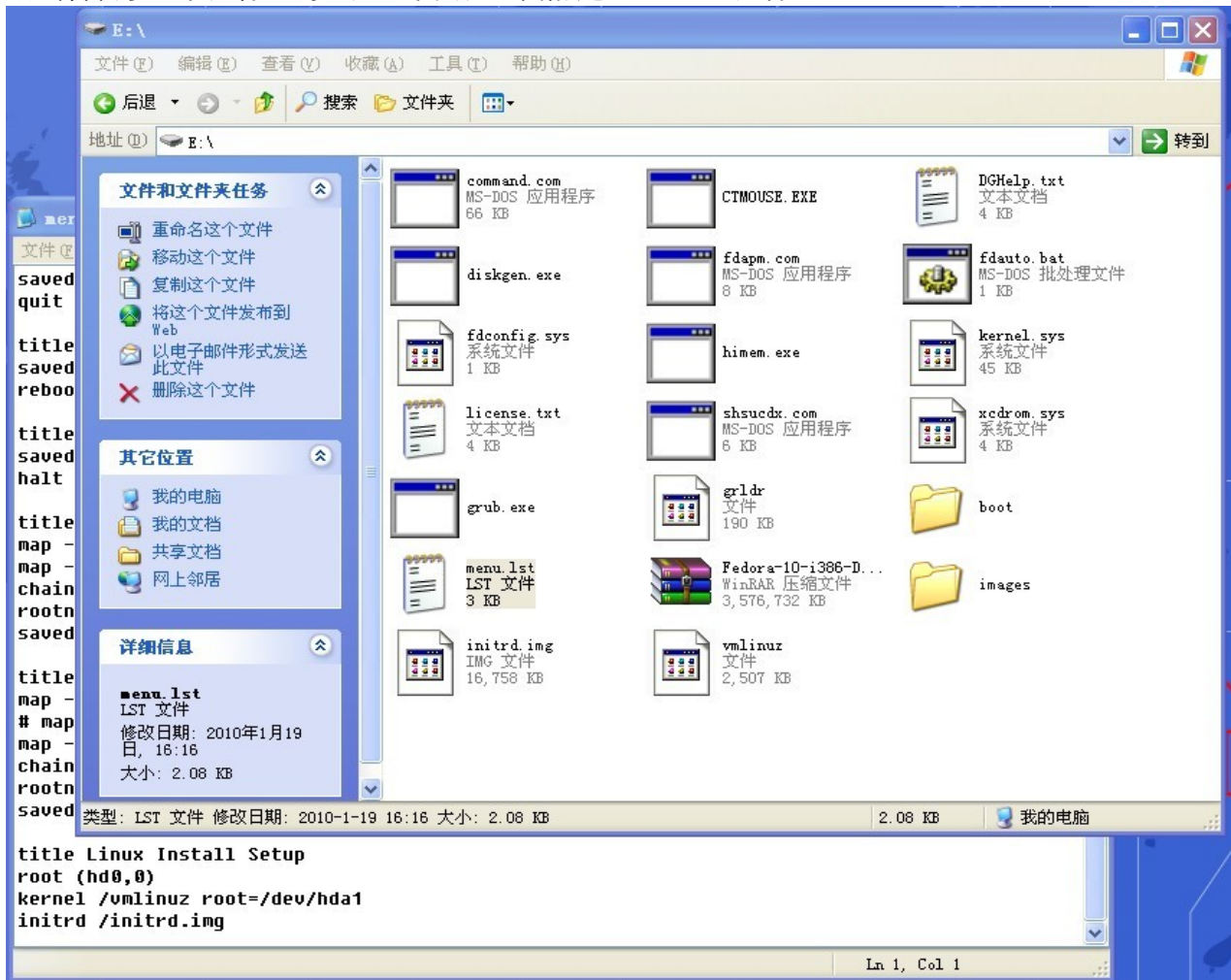
```
/dev/sda3
```

```
/dev/sdb1 -----u 盘位置，选择
```

顺便说一下，如果有配置服务器的话也可以用下面三种方式安装，这在装多台机器的时候能体会到便利之处。

再往下就是正常安装系统拉。呵呵，我的啰嗦到此结束，如有错误及不明白之处可以联系我 changzi100@yahoo.com.cn。

PS: 所装的系统为 CentOS, 如果装 Fedora 时要将 image 文件夹解压到 U 盘的根目录中。
我试着装了一下, 有些调整, 如下图, 后面的是 menu.list 文件:



Python 在数据库测试中的应用探索

Python(派森)语言是一种面向对象的用途非常广泛的编程语言, 具有非常清晰的语法特点, 适用于多种操作系统, 可以在 Windows 和 Unix 这样的系统中运行。目前在国际上非常流行, 正在得到越来越多的应用。Python 可以完成许多任务, 功能非常强大。

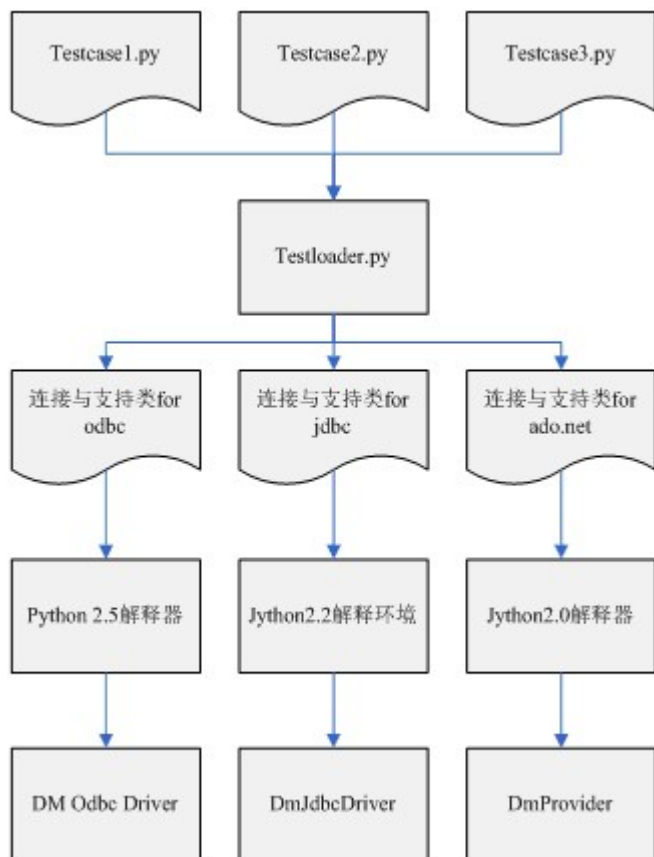
对于我们的测试工作而言, Python 最吸引我们的特性有如下几个方面:

- 1 具备语言粘合剂的能力
- 2 解释执行的机制
- 3 语法简单易学
- 4 相对较高的性能

语言粘合剂是比较形象的说法, 具体的说, Python 支持通过引入自带的 ctypes 库, 达到在 python 脚本中执行已有的动态库中的代码的目标。因此, DM 对外提供的基于 c/c++ 的接口都可以通过这个方法得到访问, 如果使用 Python 作为接口测试的载体, 得到的测试用将比使用编译型语言的测试程序更易于修改调试。

以上是针对 c/c++ 接口而言，对于 .net 平台和 JAVA 平台，我们可以分别使用 Python for .Net 和 Jython。前者使得 Python 脚本可以直接调用 CLR 代码，而后者使得 Python 可以直接调用任何的 JAVA 类库。

相信至此，使用 Python 作为跨越语言的测试工具的优越性，已经十分明显了：对于功能与性能测试，我们只需要自己编写一个解释器调用脚本，就可以只需要维护一套测试脚本，而同时可以在任何驱动接口上进行测试，构思大体如下图所示：



在上面这个图中，测试工程师只需要关心最上层的 Testcases.py 就可以，他们所写的用例可以是纯粹的测试逻辑而不需关心不同接口的实现方法，例如一个简单的测试脚本可以是下面这样的：

Testcase1.py:

```
from dbop import * #这里的 dbop 就是 odbc 版本的连接与支持类
from dmServer import * #dmServer 类用于对被测试的服务器进行启动、停止、配置等操作
from ctypes import *
rc = -10;
dmserver1 = dmServer('d:\\dmdbms\\bin\\dmserver.exe','win','12345','service',0);
dmserver1.runserver();

fb1 = dbop();
rc = fb1.connect('127.0.0.1','SYSDBA','SYSDBA','12345','SYSTEM');
rc = fb1.exec_sql_ignore("drop table aa;");#执行 drop 语句而不关心是否执行成功
rc = fb1.exec_sql("create table aa(c int);");#执行 create 语句且预期成功
```

```
rc = tb1.exec_sql_failed("create table aa(c int);");#再次执行该语句，预期失败
k=((1,1),(1,1));
```

#执行 select 且预期结果集与上面定义的 k 相同

```
rc = tb1.exec_sql_with_result("select top 2 name,id from systables;",k);
```

```
rc = tb1.disconnect();
```

```
dmserver1.stopserver();
```

可以看到，如上所示，编写用例的人员完全不需关心

connect, exec_sql_ignore, tb1.exec_sql_failed 此类的操作细节如何实现，他们只要使用并给出测试的逻辑就可以了。而测试开发人员则要实现例如上面这几个接口的 odbc 版本、jdbc 版本、和 ado.net 版本。

位于测例与支持库之间的是一个 Testloader 程序，这个程序完成这几件事情：

- 1、 读取根据要执行的测试的配置文件，确定解释器的位置和将要使用哪种驱动；
- 2、 读取测试用例的内容，并将对支持类的引用改为对应驱动，比如上面的脚本中的：
from dbop import * 可能会被改为：from dbop_jdbc import *
- 3、 调用对应的解释器执行这个测试用例。

一个比较简单的例子如下：

cfg.ini:

```
py_path=D:\ACTIVE Python\python.exe
```

```
jy_path=C:\jython2.2.1\dmjython.bat
```

```
ny_path=C:\pythonnet-2.0-alpha2\pythonnet-2.0-alpha2\python2.5-UCS2\python.exe
```

```
#driver =dotnet
```

```
driver =odbc
```

```
#driver =jdbc
```

Testloader:

```
import os
```

```
import string
```

```
import sys
```

```
def exeascript(sc,driver):
```

```
    read_script(sc)
```

```
    replace_lib_include(sc,driver)
```

```
    write_temp_script();
```

```
    cmd = get_executer_path()+ "" + "temp.py"
```

```
    newfile = file("temp.py","w")
```

```
    newfile.write(all_text);
```

```
    newfile.close()
```

```
    os.system(cmd);
```

```
    os.remove("temp.py")
```

```
mode =""
```

```
scriptfile=""
```

```
lstfile=""
```

```
arg = sys.argv[1]
```



```
tempdriver=sys.argv[2]
if string.upper(arg[arg.rfind('.')+1:len(arg)])=="PY":
    mode="py"
    scriptfile = arg
elif string.upper(arg[arg.rfind('.')+1:len(arg)])=="LST":
    mode="lst"
    lstfile=arg

jyp="";
nyp="";
cyp="";
ininame = "cfg.ini"
dataList=[]
caseList=[]
gbldriver="odbc"
readini(ininame)

for line in dataList:
    getini();
if mode=="lst":
    #to loop execute testcase
elif mode=="py":
    exeascript(scriptfile,tempdriver)
else:
    print "Not support"
```

借助上面的这个脚本，测试执行人员执行用例只需要用 D:\py\testloader>testloader.py Testcase1.py jdbc 这样类似的格式就可以自己指定驱动来执行测试用例了。如果有大量的用例，可以将这种命令写成批处理文件来执行。而不需要人工干预。

从上面的可以看到，对于功能或者单项 SQL 性能，基于 Python 的测试用例可以提供平台无关的简单的脚本编写环境，同时由于 Python 本身是一门完整的脚本语言，测例编写人员自己也可以在测试用例中编写出各种结构的程序，正所谓可繁可简。

对于各种驱动接口，Python 来编写测试用例的好处是：由于 Python 不需要编译，你所执行的也就是你所编写的，当发生异常的时候，你无须打开集成开发环境，加载测试工程、并调试，你能够很方便的看到 python 测试脚本的内容，什么地方出了异常可以立刻发现，例如：

```
from ctypes import *
rc = c_int(-12345);
dll = windll.LoadLibrary("dmodbc.dll");#加载被测试组件
#=====#
SQLHANDLE_env = pointer(c_long(0));
SQLHANDLE_cnn = pointer(c_long(0));
SQLHANDLE_stmt = pointer(c_long(0));

pdns = c_char_p("FASTDB");
```

```
puid = c_char_p("SYSDBA");
ppwd = c_char_p("SYSDBA");

#env handle
rc = dll.SQLAllocHandle(1, None, byref(SQLHANDLE_env));
print "result of henv handle alloc :%d" %rc;
#cnv handle
rc = dll.SQLAllocHandle(2, SQLHANDLE_env, byref(SQLHANDLE_cnn));
print "result of cnn handle alloc :%d" %rc;
#connect!
rc = dll.SQLConnect(SQLHANDLE_cnn, pdns, -3, puid, -3, ppwd, -3)
print "result of connect :%d" %rc;
#stmt handle
rc = dll.SQLAllocHandle(3, SQLHANDLE_cnn, byref(SQLHANDLE_stmt));
print "result of stmt handle alloc:%d" %rc;
#exec
rc = dll.SQLExecDirect(SQLHANDLE_stmt, "insert into t values(1)", -3);
print "result of exec:%d" %rc;
#free=====
rc = dll.SQLFreeHandle(3, SQLHANDLE_stmt);
print rc;
rc = dll.SQLDisconnect(SQLHANDLE_cnn);
print rc;
rc = dll.SQLFreeHandle(2, SQLHANDLE_cnn);
print rc;
rc = dll.SQLFreeHandle(1, SQLHANDLE_env);
print rc;
```

在上面我们可以看到，Python 调用 c/c++ 接口是十分容易的，只需要把动态库加载进来，然后把动态库当作一个对象实例来使用就可以了。下面将是一个使用 ado.net 接口的例子：

```
#try to use ado.net connect to database.
import System;
from Dm import * #Dm 是 DMDBMS 提供的 ado.Net 的 DataProvider
#print dir(Dm.DmCommand);
i = 0;
cnn = Dm.DmConnection("server = 127.0.0.1; User ID = SYSDBA; PWD = SYSDBA; Database =
SYSTEM; port = 12345");
cmd = Dm.DmCommand();
cmd.Connection = cnn;
cmd.CommandText = "insert into t values(1)";
cnn.Open();
i = cmd.ExecuteNonQuery();
print i;
cmd.Dispose();
```

cnn.Close();

可以看到，.net 对象的使用与在 VisualStudio 上进行开发几乎没有任何区别。通过使用 Python 进行测试用例的开发，最大的好处莫过于：学习成本非常低，测试工程师只需要学习 Python，对于其他语言稍有了解就可以了。同时只需要少量的测试开发工程师对 Python 测试框架进行维护。

这样的好处就是便于测试人员将精力专精在一个方向，免于“什么都会一点，但什么都不精”的情况。当然测试人员具备广阔的知识面，会使用各种常见的开发工具与平台是好事情，并且也是必要的，不过在短时间内要求迅速能够胜任大多数任务也是企业在人才培养上的期望目标。

突破 Linux 内核模块认证

ChinaUnix 网友：W.Z.T

本篇文章是讲内核安全方面的，不感兴趣的同学可绕过，文章写的比较仓促，有问题的话，欢迎指出。

1、为什么要突破模块验证

Linux 内核版本很多，升级很快，2 个内核版本中内核函数的定义可能都不一样，为了确保不一致的驱动程序导致 kernel oops，开发者加入了模块验证机制。它在加载内核模块的时候对模块进行校验，如果模块与主机的一些环境不一致，就会加载不成功。

看下面一个例子，它简单的输出当期系统中的模块列表：

```
1. #include <linux/kernel.h>
2. #include <linux/module.h>
3. #include <linux/init.h>
4. #include <linux/version.h>
5. #include <linux/string.h>
6. #include <linux/list.h>
7.
8. MODULE_LICENSE("GPL");
9. MODULE_AUTHOR("wzt");
10.
11. struct module *m = &__this_module;
12.
13. int print_module_test(void)
14. {
15.     struct module *mod;
16.
```

```
17.    list_for_each_entry(mod, &m->list, list) {
18.        printk("%s\n", mod->name);
19.    }
20.    return NULL;
21.}
22.
23.static int list_print_init(void)
24.{
25.    printk("load list_print module.\n");
26.
27.    print_module_test();
28.
29.    return 0;
30.}
31.
32.static void list_print_exit(void)
33.{
34.    printk("unload list_print module.\n");
35.}
36.
37.module_init(list_print_init);
38.module_exit(list_print_exit);
```

我们在 centos5.3 环境中编译一下：

1. [root@localhost list]# uname -a
2. Linux localhost.localdomain 2.6.18-128.el5 #1 SMP Wed Jan 21 10:44:23 EST 2009 i686 i686 i386 GNU/Linux

然后拷贝到另一台主机 centos5.1xen 上：

1. [root@localhost ~]# uname -a
2. Linux localhost.localdomain 2.6.18-53.el5xen #1 SMP Mon Nov 12 03:26:12 EST 2007 i686 i686 i386 GNU/Linux

用 insmod 加载：

1. [root@localhost ~]# insmod list.ko
2. insmod: error inserting 'list.ko': -1 Invalid module format

报错了，在看下 dmesg 的信息：

1. [root@localhost ~]# dmesg | tail -n 1
2. list: disagrees about version of symbol struct_module

先不管这是什么，总之我们的模块在另一台 2.6.18 的主机中加载失败。通常的做法是要在主机中对源代码进行编译，然后才能加载成功，但是如果主机中缺少内核编译环境的话，我们的 rootkit 就不能编译，也不能安装在主机之中，这是多么尴尬的事情。没错，这就是 linux kernel 开发的特点，你别指望像 windows 驱动一样，编译一个驱动，然后可以满世界去装^_^。一些 rootkit 开发者抛弃了 lkm 类型 rk 的开发，转而去打 kmem, mem 的注意，像 sk, moodnt 这样的 rk 大家都喜欢，可以在用户层下动态 patch 内核，不需要编译环境，wget 下来，install 即可。但是它也有很多缺点，比如很不稳定，而且在 2.6.x 后内核已经取消了 kmem 这个设备，mem 文件也做了映射和读写的限制。rk 开发者没法继续 sk 的神话了。反过来，如果我们的 lkm 后门不需要编译环境，也可以达到直接 insmod 的目的，这是件多么美好的事情，而且 lkm 后门更加稳定，还不用像 sk 在内核中添加了很多自己的数据结构。

2、内核是怎么实现的

我们去看看内核在加载模块的时候都干了什么，或许我们可以发现点 bug，然后做点手脚，欺骗过去：) grep 下 dmesg 里的关键字，看看它在哪个文件中：

1. [root@localhost linux-2.6.18]# grep -r -i 'disagrees about' kernel/
2. kernel/module.c: printk("%s: disagrees about version of symbol %s\n",

2.6.18/kernel/module.c:insmod 调用了 sys_init_module 这个系统调用，然后进入 load_module 这个主函数，它解析 elf 格式的 ko 文件，然后加载到内核中：

1. /* Allocate and load the module: note that size of section 0 is always
2. zero, and we rely on this for optional sections. */
3. static struct module *load_module(void __user *umod,
4. unsigned long len,
5. const char __user *uargs)
6. {
7. ...
8. if (!check_modstruct_version(sechdrs, versindex, mod)) {
9. err = -ENOEXEC;
10. goto free_hdr;
11. }
- 12.

```
13.   modmagic = get_modinfo(sechdrs, infoindex, "vermagic");
14.   /* This is allowed: modprobe --force will invalidate it. */
15.   if (!modmagic) {
16.       add_taint(TAINT_FORCED_MODULE);
17.       printk(KERN_WARNING "%s: no version magic, tainting kernel.\n",
18.             mod->name);
19.   } else if (!same_magic(modmagic, vermagic)) {
20.       printk(KERN_ERR "%s: version magic '%s' should be '%s'\n",
21.             mod->name, modmagic, vermagic);
22.       err = -ENOEXEC;
23.       goto free_hdr;
24.   }
25....
26.}
```

check_modstruct_version 就是用来计算模块符号的一些 crc 值，不相同就会出现我们在 dmesg 里看到的 “disagrees about version of symbol” 信息。get_modinfo 取得了内核本身的 vermagic 值，然后用 same_magic 函数和内核的 vermagic 去比较，不同也会使内核加载失败。所以在这里，我们看到内核对模块验证的时候采用了 2 层验证的方法：模块 crc 值和 vermagic 检查。

继续跟踪 check_modstruct_version，现在的内核默认的都开启了 CONFIG_MODVERSIONS，如果没有指定这个选项，函数为空，我们的目的是要在 As, Centos 下安装模块，redhat 不是吃干饭的，当然开了 MODVERSIONS 选项。

```
1. static inline int check_modstruct_version(Elf_Shdr *sechdrs,
2.             unsigned int versindex,
3.             struct module *mod)
4. {
5.     const unsigned long *crc;
6.     struct module *owner;
7.
8.     if (!__find_symbol("struct_module", &owner, &crc, 1))
9.         BUG();
10.    return check_version(sechdrs, versindex, "struct_module", mod,
```

```
11.         crc);  
12.}
```

__find_symbol 找到了 struct_module 这个符号的 crc 值，然后调用 check_version 去校验：

```
1. static int check_version(Elf_Shdr *sechdrs,  
2.         unsigned int versindex,  
3.         const char *symname,  
4.         struct module *mod,  
5.         const unsigned long *crc)  
6. {  
7.     unsigned int i, num_versions;  
8.     struct modversion_info *versions;  
9.  
10.    /* Exporting module didn't supply crcs? OK, we're already tainted. */  
11.    if (!crc)  
12.        return 1;  
13.  
14.    versions = (void *) sechdrs[versindex].sh_addr;  
15.    num_versions = sechdrs[versindex].sh_size  
16.        / sizeof(struct modversion_info);  
17.  
18.    for (i = 0; i < num_versions; i++) {  
19.        if (strcmp(versions[i].name, symname) != 0)  
20.            continue;  
21.  
22.        if (versions[i].crc == *crc)  
23.            return 1;  
24.        printk("%s: disagrees about version of symbol %s\n",  
25.            mod->name, symname);  
26.        DEBUGP("Found checksum %IX vs module %IX\n",  
27.            *crc, versions[i].crc);  
28.        return 0;
```

```
29.     }
30.     /* Not in module's version table. OK, but that taints the kernel. */
31.     if (!(tainted & TAINTE_FORCED_MODULE)) {
32.         printk("%s: no version for \"%s\" found: kernel tainted.\n",
33.             mod->name, symname);
34.         add_taint(TAINTE_FORCED_MODULE);
35.     }
36.     return 1;
37.}
```

它搜寻 elf 的 versions 小节，循环遍历数组中的每个符号表，找到 struct_module 这个符号，然后去比较 crc 的值。现在有个疑问，versions 小节是怎么链接到模块的 elf 文件中去的呢？在下编译后的生成文件，有一个 list.mod.c

```
1. [root@localhost list]# cat list.mod.c
2. #include <linux/module.h>
3. #include <linux/vermagic.h>
4. #include <linux/compiler.h>
5.
6. MODULE_INFO(vermagic, VERMAGIC_STRING);
7.
8. struct module __this_module
9. __attribute__((section(".gnu.linkonce.this_module"))) = {
10. .name = KBUILD_MODNAME,
11. .init = init_module,
12. #ifdef CONFIG_MODULE_UNLOAD
13. .exit = cleanup_module,
14. #endif
15. };
16.
17. static const struct modversion_info ____versions[]
18. __attribute_used__
19. __attribute__((section("__versions"))) = {
20.     { 0x89e24b9c, "struct_module" },
```



```
21.    { 0x1b7d4074, "printk" },
22.};
23.
24.static const char __module_depends[]
25.__attribute_used__
26.__attribute__((section(".modinfo"))) =
27."depends=";
28.
29.
30.MODULE_INFO(srcversion, "26DB52D8A56205333D414B9");
```

这个文件是模块在编译的时候，调用了 linux-2.6.18/scripts/modpost 这个文件生成的。里面增加了 2 个小节.gnu.linkonce.this_module 和__versions。__versions 小节的内容就是一些字符串和值组成的数组，check_version 就是解析这个小节去做验证。这里还有一个 MODULE_INFO 宏用来生成模块的 magic 字符串，这个在以后的 vermagic 中要做验证。

先看下 vermagic 的格式：

```
1. [root@localhost list]# modinfo list.ko
2. filename:    list.ko
3. author:     wzt
4. license:    GPL
5. srcversion:  26DB52D8A56205333D414B9
6. depends:
7. vermagic:    2.6.18-128.el5 SMP mod_unload 686 REGPARM 4KSTACKS gcc-4.1
```

这里可以看到 vermagic 跟内核版本，smp，gcc 版本，内核堆栈大小都有关。

```
1. /* First part is kernel version, which we ignore. */
2. static inline int same_magic(const char *amagic, const char *bmagic)
3. {
4.     amagic += strcspn(amagic, "");
5.     bmagic += strcspn(bmagic, "");
6.     return strcmp(amagic, bmagic) == 0;
7. }
```

same_magic 忽略了对内核版本的判断，直接比较后面的值。

3、怎样去突破

知道了内核是怎么实现的了， 下面开始想办法绕过这些验证：)

3.1 怎么突破 crc 验证：

再仔细看下代码：

```
1.     for (i = 0; i < num_versions; i++) {
2.         if (strcmp(versions[i].name, symname) != 0)
3.             continue;
4.
5.         if (versions[i].crc == *crc)
6.             return 1;
7.         printk("%s: disagrees about version of symbol %s\n",
8.             mod->name, symname);
9.         DEBUGP("Found checksum %lX vs module %lX\n",
10.            *crc, versions[i].crc);
11.        return 0;
12.    }
13.    /* Not in module's version table. OK, but that taints the kernel. */
14.    if (!(tainted & TAINTE_FORCED_MODULE)) {
15.        printk("%s: no version for \"%s\" found: kernel tainted.\n",
16.            mod->name, symname);
17.        add_taint(TAINTE_FORCED_MODULE);
18.    }
19.    return 1;
```

check_version 在循环中只是在寻找 struct_module 符号， 如果没找到呢？ 它会直接返回 1！ 没错， 这是一个逻辑 bug， 在正常情况下， module 必会有一个 struct_module 的符号， 这是 modpost 生成的。如果我们修改 elf 文件， 把 struct_module 这个符号改名， 岂不是就可以绕过 crc 验证了吗？ 先做个实验看下：

.mod.c 是由 modpost 这个工具生成的， 它在 linux-2.6.18/scripts/Makefile.modpost 文件中被调用， 去看下：

```
1. PHONY += __modpost
2. __modpost: $(wildcard vmlinux) $(modules:.ko=.o) FORCE
3.    $(call cmd,modpost)
```

我们用一个很土的方法，就是在编译模块的时候，modpost 生成.mod.c 文件后，暂停下编译，sleep 30 秒吧，我们用这个时间去改写下.mod.c，把 struct_module 换个名字。

1. PHONY += __modpost
2. __modpost: \$(wildcard vmlinux) \$(modules:.ko=.o) FORCE
3. \$(call cmd,modpost)
4. @sleep 30

随便将 struct_module 改个名：

1. [root@localhost list]# cat list.mod.c
2. #include <linux/module.h>
3. #include <linux/vermagic.h>
4. #include <linux/compiler.h>
- 5.
6. MODULE_INFO(vermagic, VERMAGIC_STRING);
- 7.
8. struct module __this_module
9. __attribute__((section(".gnu.linkonce.this_module"))) = {
- 10..name = KBUILD_MODNAME,
- 11..init = init_module,
- 12.#ifdef CONFIG_MODULE_UNLOAD
- 13..exit = cleanup_module,
- 14.#endif
- 15.};
- 16.
- 17.static const struct modversion_info ____versions[]
- 18.__attribute_used__
- 19.__attribute__((section("__versions"))) = {
20. { 0x89e24b9c, "stauct_module" },
21. { 0x1b7d4074, "printk" },
- 22.};
- 23.
- 24.static const char __module_depends[]

```
25.__attribute_used__
26.__attribute__((section(".modinfo"))) =
27."depends=";
28.
29.MODULE_INFO(srcversion, "26DB52D8A56205333D414B9");
```

我们是在 centos5.3 下编译的，然后拷贝到 centos5.1 下，在执行下 insmod 看下：

```
1. [root@localhost ~]# insmod list.ko
2. [root@localhost ~]# dmesg | tail
3. ata_piix
4. libata
5. sd_mod
6. scsi_mod
7. ext3
8. jbd
9. ehci_hcd
10.ohci_hcd
11.uhci_hcd
```

成功了！这跟我们预期的一样，我们用这个逻辑 bug 绕过了模块的 crc 验证！这个 bug 直到 2.6.31 版本中才得到修正。我们可以用这种方法在 redhat 主机中任意安装模块了。那么怎样绕过在 2.6.31 以后的内核呢？

看下它是怎么修补的：

```
1.     for (i = 0; i < num_versions; i++) {
2.         if (strcmp(versions[i].name, symname) != 0)
3.             continue;
4.
5.         if (versions[i].crc == *crc)
6.             return 1;
7.         DEBUGP("Found checksum %IX vs module %IX\n",
8.             *crc, versions[i].crc);
9.         goto bad_version;
10.    }
11.
```

```
12.    printk(KERN_WARNING "%s: no symbol version for %s\n",
13.        mod->name, symname);
14.    return 0;
15.
16.bad_version:
17.    printk("%s: disagrees about version of symbol %s\n",
18.        mod->name, symname);
19.
20.    return 0;
```

如果没找到 struct_module 也会返回 0，这样我们就必须将 struct_module 的值改为正确后，才能继续安装。如何找到模块符号的 crc 值呢？我们可以去找目标主机中那些已被系统加载的模块的 crc 值，如 ext3 文件系统的模块，自己写个程序去解析 elf 文件，就可以得到某些符号的 crc 值了。

还有没有更简单的方法呢？去 /boot 目录下看看，symvers-2.6.18-128.el5.gz 貌似和 crc 有关，gunzip 解压后看看：

```
1. [root@localhost boot]# grep 'struct_module' symvers-2.6.18-128.el5
2. 0x89e24b9c    struct_module    vmlinux EXPORT_SYMBOL
```

原来内核中所有符号的 crc 值都保存在这个文件中。如何改写 struct_module 的值呢，可以用上面那个土方法，或者自己写程序去解析 elf 文件，然后改写其值。本文最后附上一个小程序用来修改 elf 的符号和 crc 值。

3.2 如何突破 vermagic 验证：

如果我们用 list.mod.c 中的做法，用 MODULE_INFO 宏来生成一个与目标主机相同的 vermagic 呢？答案是否定的，gcc 链接的时候会把 modinfo 小节链接在最后，加载模块的时候还是会读取第一个.modinfo 小节。

我们可以用上面那种很土的方法，先用 modinfo 命令得到目标主机中某个模块的信息：

```
1. [root@localhost list]# modinfo /lib/modules/2.6.18-128.el5/kernel/fs/ext3/ext3.ko
2. filename:    /lib/modules/2.6.18-128.el5/kernel/fs/ext3/ext3.ko
3. license:    GPL
4. description: Second Extended Filesystem with journaling extensions
5. author:     Remy Card, Stephen Tweedie, Andrew Morton, Andreas Dilger,
               Theodore Ts'o and others
6. srcversion:  B048AC103E5034604A721C5
7. depends:    jbd
8. vermagic:    2.6.18-128.el5 SMP mod_unload 686 REGPARM 4KSTACKS gcc-4.1
```


9. module_sig:

```
883f3504977495e4f3f897cd3dced211288209f551cc1da557f96ea18d9a4efd6cfb0fc  
2612e009c8845fd776c825d586f492ceab19e17b2319da8f
```

然后在用那个很土的方面，将.mod.c 中 vermagic 值进行修改。还有一种直接修改 elf 文件的方法，附录在本文后面。

4、总结

前面有一点没有提到，就是某些内核版本的相同接口的函数代码可能已经变化，这样在使用这项技术的时候，最好在同一个大内核版本使用。你也可能感觉要想跨平台安装模块有些麻烦，这里还有 2 个方法，作为一个专业搞渗透的人来说，他会自己在本地装很多发行版本的 linux，特别是 root 掉一台主机后，会在本地装一个一模一样的发行版本，smp、kernel stack size、gcc version 都一样。在本地机器装上开发环境，这样编译出来的模块也是可以直接装到目标主机上的，但这很麻烦，因为 linux 有太多的发行版本了：），另一个方法就是自己装一个 linux，编译下内核，然后将 build 后的开发包集成到自己的后门里，压缩后大概几 m。然后传到主机去解压，编译。庆幸的是，现在大多数主机中都有内核开发环境，直接去主机编译就 ok 了。

5、参考

【1】 linux kernel source code

<http://www.kernel.org>

【2】 module injection in 2.6 kernel - Coolq

[http://www.nsfocus.net/index.php ... o=view&mid=2533](http://www.nsfocus.net/index.php...o=view&mid=2533)

git 几种应用模式

ChinaUnix 网友: blumdrum

git 是新出现的分布式源码控制系统,分布式(可离线操作),速度快,操作简单,在很多场合开始做 GIT,象 LINUX 内核,QT,Andriod 都采用 git 来做源码控制.现在 GIT 的文档很多,但对于初学者和一般开发者,对 GIT 的使用场景没有系统的介绍.因此有了这一篇文章。

本文假设阅读者有 GIT 使用常识,和源码控制基本概念.

GIT 的核心概念是用在开发者本地建立一个.git 仓库,所有的操作都是首先在这个本地.git 进行提交.然后再向其它位置传输.在本地创建仓库 用如下命令,后面不再重复.

git init <仓库名>

这里总结五种 GIT 应用模式,其中一二模式用单人或小团队,三,四用于大团队或集中管理,五.用于大团队或互联网。

一.GIT 单人应用模式.

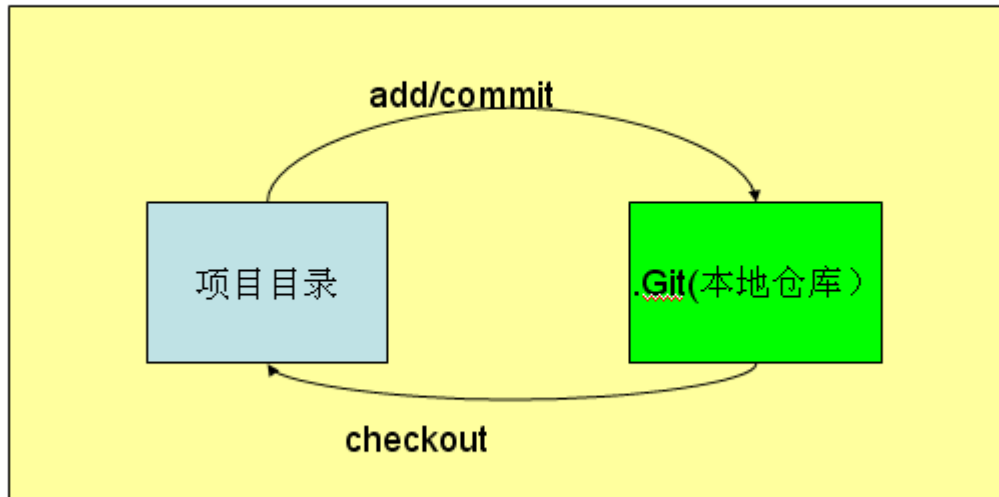
这种模式常用于单人开发小软件,或者个人文档,源码的保存.这种处理也比较简单.

在项目开始时.要用 git add <文件名> 把需要提交的文件加入.git 仓库

然后在项目中用 git commit 进行提交备份.

用 `git checkout <tag>` 取出某一个备份

GIT单人模式

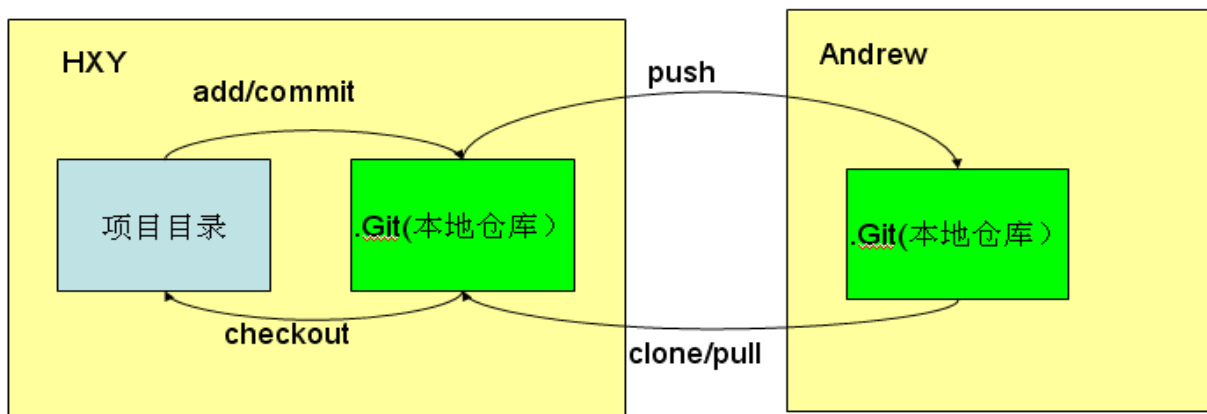


bluedrum@163.com

二.多人对等开发模式

用于多人处理平等地位,但是没有服务器的比较松散的情况,比如开源软件开发,公司团队较少没有使用服务器的情况.在这种模式下,一个开发者可以用 `git clone` 命令完整从外一个开发者下载项目。在项目开发中,开发者可以用 `git pull` 从对方机器更新数据,或者用 `git push` 把向对方机器传输数据。

GIT多人对等模式



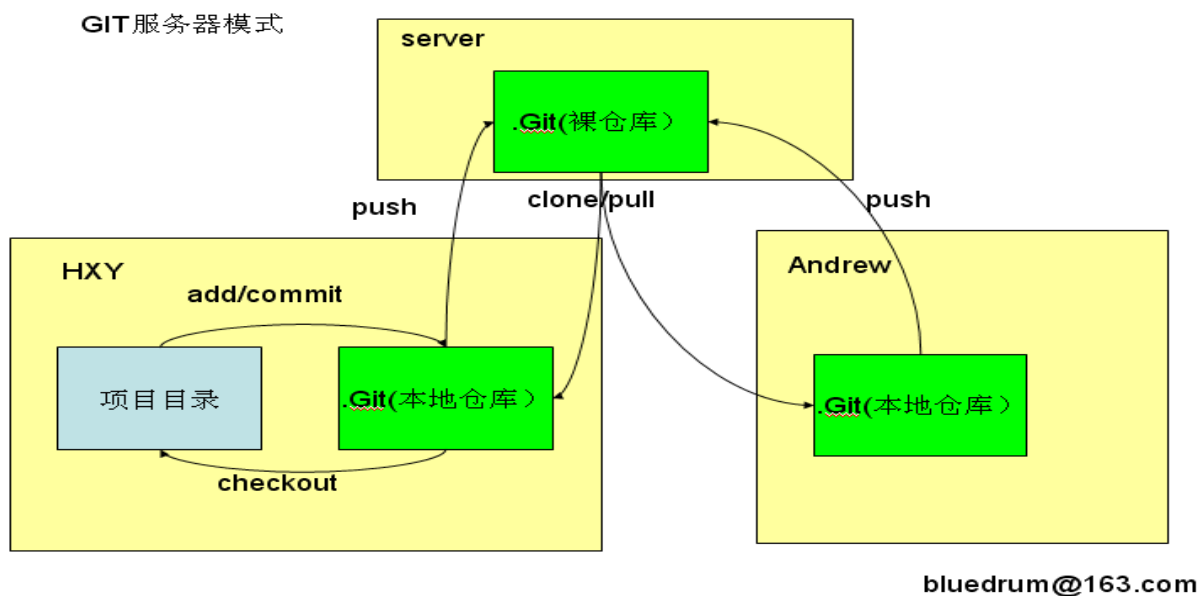
bluedrum@163.com

三.Git 服务器模式

适合一个公司有集中服务器的情况,首先需要在服务器建立一个.git 裸仓库(意思只有版本数据,没有临时文件),建立方法,在服务器端:

`git --bare init --shared`

或者用 `git clone --bare` 克隆一个仓库,然后各个开发者可以用 `git pull/clone` 来下载数据,用 `git push` 来向服务器提交数据。

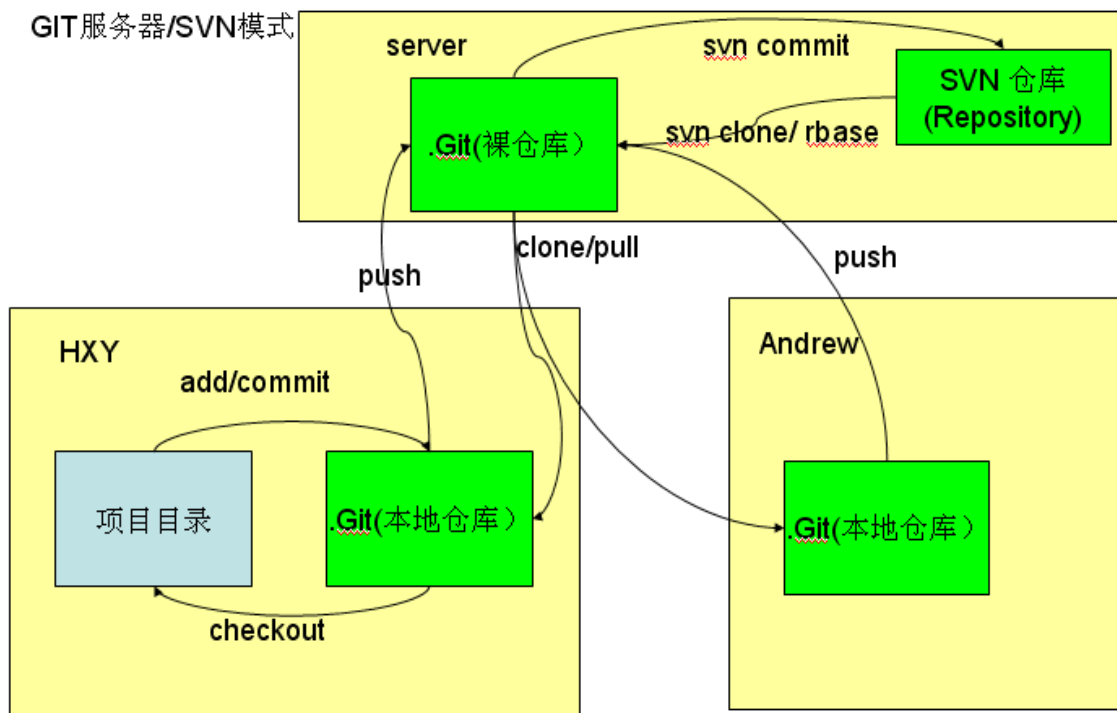


四与 SVN 服务器配合模式

相对于 git,传统的 SVN 有完善的权限管理和用户认证。并支持多用户并发。很多公司原来也是采用 SVN 的。并且与 SVN 配合的软件也相当多。因此在保留 SVN 服务器的同时。使用 git 来提交也是比较好的模式。

在 git 的新版本已经集成与 svn 的操作。使用 `git-svn`

- `git-svn svn clone` 从 SVN 服务器复制到.git 仓库
- `git-svn rebase` 从 svn 服务器更新数据到.git 仓库
- `git-svn dcommit` 向 svn 服务器提交数据

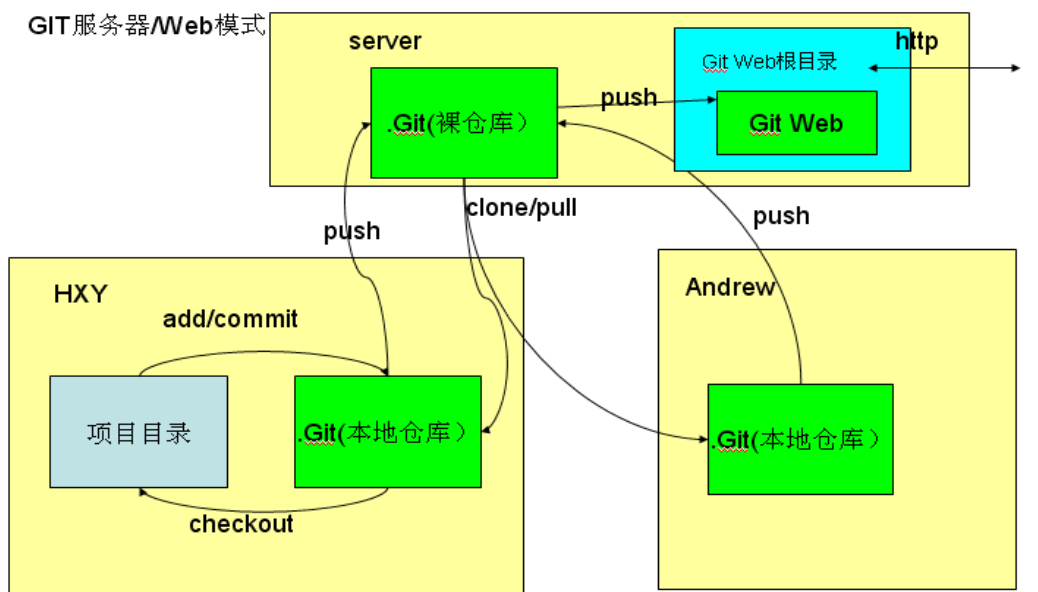


bluedrum@163.com

五 git 服务器与 Web 服务器集成

这种模式可以把 git 的内容发布到支持 cgi 的 WEB 服务器上，如 Apache/httpd 等。

样例：<http://git.kernel.org/>



bluedrum@163.com

Kernel 花哨的编译连接技术

ChinaUnix 网友: hyl

阅读 linux 的入门门槛在逐步提高, 看看这些技术的应用吧.

<http://www.ohse.de/uwe/articles/gcc-attributes.html>

我最先知道的编译连接是 `pack(0), #define __packed __attribute__((packed))`
因为做网络通讯, 这个很常见. 用于将对齐改为 1byte, 就是不要在结构内增加 pad 的内容. 可是打开 linux 源码看看, `tcphdr` 竟然没有 `packed` 属性, 真是雷阿。

(下面默认是 32bit 平台)

`make V=1 net/ipv4/udp.o`, 无线索. 没有 `-fpack-struct`, 没有类似 VC 的 `-Zpn...` 电话召唤 wheel, 一起试验, 发现自己还是需要补充基础知识.

```
struct udphdr {
    __be16  source;
    __be16  dest;
    __be16  len;
    __sum16  check;
};
```

写程序, 打印 `sizeof udphdr`, 可以发现这种声明是么有 pad 的. 规则是: `char char short; short short long; long long (long long)`.

这个规则在各种平台(应该)通用. (否则 mips 上岂不是也要有问题阿)

令人惊奇的是: 下面的声明, `sizeof` 值是 1, 用做数组就会效率不高的. (一下在 x86-32 上验证, mips32 上肯定不同, 未试,)

```
struct udphdr {
    char x;
};
```

而这个 `sizeof` 是 8

```
struct udphdr {
    char x;
    long x
};
```

最后这个 `sizeof` 是 4

```
struct udphdr {
    char x;
    short x
};
```


结论: 最可考的做法是测试下.

这篇可以参考下:

<http://hi.baidu.com/phps/blog/item/f03eb93ee12f49fa838b1365.html>

kernel image 加载的逻辑地址, kernel image 各段的起始/结束地址 init 函数的收集, 这个都能在 lds 中找到. 不幸, lds 文件找不到了....

```
find -iname *.lds
./arch/x86/boot/compressed/vmlinux_32.lds
./arch/x86/boot/compressed/vmlinux_64.lds
./arch/sh/boot/compressed/vmlinux_64.lds
./arch/mn10300/boot/compressed/vmlinux.lds
./arch/alpha/boot/bootloader.lds
./arch/ia64/hp/sim/boot/bootloader.lds
./arch/ia64/scripts/check-segrel.lds
./arch/ia64/module.lds
./arch/arm/boot/bootp/bootp.lds
./arch/m68k/kernel/vmlinux-std.lds
./arch/m68k/kernel/vmlinux-sun3.lds
./arch/m68k/kernel/module.lds
./arch/h8300/boot/compressed/vmlinux.lds
```

ft, 仔细看看 arch/kernle/目录, 原来变成了.lds.S, ft.

SECTIONS

```
{
. = LOAD_OFFSET + LOAD_PHYSICAL_ADDR;
.....
.initcall.init : AT(ADDR(.initcall.init) - LOAD_OFFSET) {
    __initcall_start = .;
    INITCALLS
    __initcall_end = .;
}

}
```

好处是, 能够接受 config 配置的宏. 更灵活.

汇编文件手动定义调试信息, 参考 dwarf2.h.

采用宏来生成 static inline 函数, 这样, 一些函数的定义你有可能搜不到. 比如 x86 平台的 readb writeb: x86/io.h

```
#define build_mmio_read(name, size, type, reg, barrier) \
static inline type name(const volatile void __iomem *addr) \
{ type ret; asm volatile("mov" size " %1,%0":reg (ret) \
:"m" (*(volatile type __force *)addr) barrier); return ret; }
```

```
build_mmio_read(readb, "b", unsigned char, "=q", : "memory")
build_mmio_read(readw, "w", unsigned short, "=r", : "memory")
build_mmio_read(readl, "l", unsigned int, "=r", : "memory")
```

采用 WEAK 符号

汇编中采用 WEAK 宏, 定义在 linux/linkage.h

```
#ifndef WEAK
#define WEAK(name)    \
    .weak name;    \
    name:
#endif
```

C 中采用 attribute : linux/compiler-gcc.h

```
#define __weak      __attribute__((weak))
```

在 mips/kernle/timer.c 中有个 weak 符号, 用于提供一个 dummy 的实现,

```
int __weak rtc_mips_set_time(unsigned long sec)
{
    return 0;
}
```

在 mips/sgi-ip22/ip22-time.c 中可以提供一个非 weak 的符号覆盖 weak 符号, 用于提供特定平台的实现.

```
int rtc_mips_set_time(unsigned long tim)
{
    struct rtc_time tm;
    unsigned int save_control;
    unsigned long flags;

    rtc_time_to_tm(tim, &tm);

    tm.tm_mon += 1;    /* tm_mon starts at zero */
    tm.tm_year -= 40;

    ....
}
```

然后, weak 还可以和 alias 配合, 嘿嘿, 门槛高阿.

发现一个有意思的属性: linux/compiler-gcc.h

```
#define __naked          __attribute__((naked)) notrace  
naked
```

Found in versions: 3.0-3.4 Description:

Use this attribute on the ARM, AVR, C4x and IP2K ports to indicate that the specified function **does not need prologue/epilogue** sequences generated by the compiler. It is up to the programmer to provide these sequences.

网友热评

热点技术评论

[Linux 有那些看代码工具?](#)
[关于 DNS \(bind9\) 的递归问题](#)
[postgresql 怎么查某个正执行的 SQL 语句啊?](#)
[grep 能否抓 2 行?](#)
[求“警告：比较指针和整数”的解决方法](#)
[关于服务器安装 sco unix 5.05](#)
[awk 中获取某字符串中有几个空格](#)
[关于 startserver 显示的项目内容](#)
[使用 DBI 连接 Oracle 数据库的问题](#)
[文件 read 和 write 是否会引起线程切换?](#)
[关于 bakbone 恢复 oracle](#)
[perl 水平如何提升](#)
[C++模板无法实现在运行时的泛型功能!](#)
[关于 Mysql InnoDB 性能](#)
[mysql cluster 纯属招摇撞骗](#)
[perl 如何对 2G 的大数量文本文件内容排序?](#)
[怎样用命令方式打开一个新的终端窗口](#)
[unix 下对文件的列值排序比较问题](#)
[怎么产生 n 个不重复等概率随机数?](#)
[sybase 索引数据物理存储结构分析](#)
[FREEBSD 8 常用命令](#)
[向 socket 一次写多少数据合适?](#)
[面试中八大经典问题的对应方法](#)
[请教用 awk 求最大值的问题](#)
[char \(* test\[4\]\)\[7\] 定义了什么东西?](#)
[国外空间商为什么只用 linux 而不用 bsd](#)
[龙芯笔记本安装 debian6 和 android 双系统](#)

热点新闻评论

[建行个人网银率先全面支持 Firefox 浏览器](#)
[bsd 有精神文化和代表没?](#)
[c++ 的终结者?](#)
[我终于还是买了 win7 正版](#)
[Gnome 和 Mac 只差一个主题](#)
[民生银行系统瘫痪数小时 影响涉及全国](#)
[google 为什么用 linux 而不用 bsd 呢](#)
[PostgreSQL 大规模并发情况下性能如何?](#)
[微软打击盗版有新招儿](#)
[大家都在使用那个发行版做桌面啊](#)
[有多少人在项目组里有决定权?](#)
[龙芯可否代工生产?](#)
[中国做技术没前途](#)
[PC-BSD 8.0 正式版发布](#)
[MySQL 真的是不堪大用么?](#)
[Java 的可移植性比 C 好纯属扯淡](#)
[FreeBSD 搞个桌面挺麻烦啊](#)
[请问大家都用 bsd 做什么?](#)
[李开复：Vista 系统失败的真正内幕](#)
[NetBSD 主要用途是什么](#)
[现在高校学 linux 的学生是否越来越少了?](#)
[NBU 7 的产品发布 PPT 下载!](#)
[26 了，往 linux 方面发展还是坚持学 C++](#)
[如何成长为顶级 linux 系统维护?](#)
[C++的 cout 就是个杯具](#)
[麻烦靠 perl 谋生的前辈们进来给点建议](#)
[为什么 C 语言接近硬件?](#)